COLLABORATIVE LEARNING TOOLS

(MAIN MODULE)

CHIN WAN SIN

WEK010048

SOFTWARE ENGINEERING DEPARTMENT

SUPERVISOR: PUAN ZARINAH MOHD. KASIRUN

MODERATOR: PUAN AZWINA MOHD. YUSOF

ACKNOWLEDGEMENTS

I would like to express my appreciation to the several people who have contributed to the successful completion of my thesis report. First of all, I would like to express my gratitude to my supervisor, Pn. Zarinah Mohd Kasirun. She had spent a lot of her precious time on supervised me and gave me guidance and advises especially when I was facing problems with my report writing. Secondly, I would like to thank Pn. Azwina Mohd Yusof, my moderator, for her valuable consideration during my VIVA presentation. Without this, I would not realize the needs of refining my project design. Besides, I would also like to thank my supervisor and moderator for willing to answer my survey questionnaire.

I would also like to take this opportunity to make a special thank to my group members, Teh Hwee See and Teo Poh Ling. I wouldn't be easily to go through all the obstacles without their supports and kindly helps. Thanks also to my housemates and friends for encouraging me for the past few months.

My special thank is due to all the respondents that has spent their precious time to answer my survey questionnaire and interviews. I greatly appreciate the staffs from faculty of Educational in University Malaya. Their willing and effort to help us to collect the survey form from the lecturers will never be forgotten.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Project Overview

With the rapid change of technological, it is not astonishing that the teaching and learning approach is also having its evolution era. In fact, traditional methods are not enough to support or fulfill nowadays-educational world anymore. Thus, through the studies and researches result show that collaborative learning is one of the suggested teaching and learning approach.

What is collaborative learning? Collaborative learning is an educational approach to teaching and learning that involves groups of students working together to solve a problem, complete a task, or create a product. It is based on the idea that learning is a naturally social act in which the participants talk among themselves *[Gerlach, 1994]*. In a collaborative learning setting, students have the opportunity to converse with peers, present and defend ideas, exchange diverse beliefs, question other conceptual frameworks, and are actively engaged *[Smith and MacGregor, 1992]*.

Collaborative learning is a slow and tedious process. It requires all the participants to be present at a particular place at the same time for a discussion purpose. Therefore, the potential of collaborative learning was still not fully developed. This brings us the idea of developing web-based collaborative learning tools called Collaborative Learning Tools (CLT).

CLT focuses on two collaborative learning methods: Jigsaw Puzzle (developed by Teo Poh Ling) and Think-Pair-Share (developed by Teh Hwee See).

While this project main purpose is to integrate these two methods into a complete collaborative learning tool.

CLT enable large groups of students to be taught and assessed using a group based approach. Regardless of the specific approach taken or how much of the ubiquitous lecture-based course is replaced, the goal is the same: to shift learning from a teacher-centered to a student-centered model *[Smith and MacGregor, 1992]*. CLT comprises an integrated set of tools: -

- *Publishing tool* – to allow the instructor to publish discussion materials.
- *Communication tool (such as email)* – to allow communication between instructor and students.
- *Management tool* – to aid the instructor to control and manage the students, courses, announcements, discussions and events.

In vision of this project is a totally new idea; future enhancement will be focused on adding more collaborative learning methods to the system. This project vision is to construct a collaborative learning tool that tackles the requirements of a wider user base.

1.2 Motivation

Based on the information and research results from Internet, we have found that collaborative learning is an advantage-learning tool. However, the current web-based collaborative learning systems are quite expensive. Moreover, most of the available systems are focus on e-learning rather than collaborative learning. Thus, we come

out an idea to develop a web-based collaborative system. Furthermore, we found that it is also well suite with the Smart School objectives.

Smart School is one of the seven flagship applications for implementation in blueprint of Multimedia Super Corridor (MSC). It is a learning institution that has been systematically reinvented in terms of teaching-learning practices and school management. The objective is to alter the school culture to teaching-learning and management processes. The focus of the Smart School idea is as follows: -

- Emphasis on maturity of thought, application of information technology and assimilation of noble values.
- Proficiency in science and mathematics.
- Enhancement of performance according to individual capabilities.
- Contribution to the development of knowledge.

In Smart School, students, teachers and the community have their own role and responsibilities. Learning will be self-directed and provide student with varying capability while teacher will play the important role of facilitators. Therefore, we make a decision to develop web-based collaborative learning tools, which is quite well matched with the objectives of this flagship application.

1.3 Statement of Problems

Collaborative learning is a philosophy of teaching that focus on working together, building together, learning together, changing together and improving together. Moreover, it has been wide spreading into web-based service, which fit today's globalized world. However, it still has its loophole. Below is the list of its problems:

a.) Current web-based system more likely an e-learning tool

Through our research, we found that most of the current web-based collaborative learning systems pay more focus on online learning instead of playing its role as a real collaborative learning tool. For an example, www.blackboard.com only provide web-based learning environment for the users but not relevant collaborative learning techniques (such as Jigsaw Puzzle and Think-Pair-Share).

b.) Tedious and bureaucracy process

Collaborative learning process involves tedious and bureaucracy steps in preparation. These time-consuming and cumbersome works always burden instructor. Thus, instructors are always avoiding themselves from using it as their teaching method.

c.) Difficult to keep track of the progress

In manual approach, it is very hard to keep track of how the group is performing and how each member is fulfilling the assigned role. Instructor is unable to supervise well of each group's discussion process. Besides, instructor is also unable to respond to the participants for their misunderstanding of the topic given.

d.) Space limitation

Without assistance of powerful web-based technology, any purpose or proceedings, which involve teamwork or group effort, may need the attendance of all the participants in a particular location. This space limitation has lead to inflexibility and inefficiency of manual system.

e.) **Inexperience instructor on using the collaborative learning technique**

Inexperience instructor may be lost when applies the collaborative learning approach due to insufficient of guides and references. Consequently, the flow of the collaborative learning process will easy out of control.

1.4 Project Objectives

CLT main objective is to provide a complete and efficient learning environment for collaborative learning approach. Some of other vital objectives are: -

- To enhance understanding of collaborative learning.
- To provide an environment for integration of two collaborative learning methods: Jigsaw Puzzle & Think-Pair-Share.
- To stimulate these two collaborative learning methods in a complete form of tool.
- To provide a fully self-control/self-management workspace for authorized instructor to organized the collaborative learning (online).

1.5 Project Scope

The implementation of CLT will require a significant of man-hours, so the project has to be defined to ensure the system can be completed in a limitation of time. The scope for this project is limited to the Internet users from secondary education institute in Malaysia. Two main sections that involved in the CLT are depicted as below: -

- **Instructor Section**

Teacher needs to register as a member before he/she can start using the CLT. Once he/she had become authorized user, he/she is assigned for a workspace,

where he/she has a fully control to manage his/her own courses and students. He/she has to input the student's information and give them the password to login. Besides, he/she also have the access right to post an announcement, create discussion session and put on some relevant information materials to the tool, where the students can only access them without modify or remove them. He/she also can send email to the students using the tool.

- ***Students Section***

Students from secondary school that are under particular instructor and have given password can join in the certain discussion session. They can use the tool to view their records; result, announcements and discussions materials or even notes. In additionally, they can communicate with each other by sending email.

1.6 Methodology

Methodology is a software process model that represent of a software process. It can form a common understanding of the activities/tasks that involved in software development. Besides, it helps in finding the inconsistencies, redundancies and omissions in the process to ensure that the final product meets the requirements of its end-users, within a predicted budget and schedule.

The approach that will be used throughout this project is *Rational Unified Process*. It consists of four fundamental phases, which are including inception, elaboration, construction and transition.

The main purpose of choosing this model is because of it allowed iteration; where a complete development loop resulting in a release (internal or external) of an

executable product, a subset of the final product under development, which grows incrementally from iteration to iteration to become the final system. This methodology will be elaborate more thoroughly in Chapter 3.

1.7 Expected Outcome

The deliverables of the CLT is expected to meet the following features: -

- Courses and students management.
- Announcement, discussion and events on calendar management.
- Integrate the two collaborative learning methods as well.
- Access the ongoing discussion content.

CLT is also expected to meet the non-functional performance: -

- ***High reliability and security***
  - The system is expected to perform its intended function with required precision and accuracy.
  - The system should be equipped with sufficient security features to enhance its reliability.
  - Each access by different users group should be authenticated and validated by the system as well.
  - It should be able to operate all the time.

- *Efficient*
  - The system is expected to make a fully use of system resources including the memory utilization and processing time.

- ***Portable***
  - The system should be able to operate in multiple platforms.

- ***Usable***
  - The system must be easy to use, especially to the two mainly users group, instructors and students as we define early in the project scope.
  - Interfaces should be user friendly, intuitive and consistent.

- ***Easy to maintain and enhance***
  - The system is developed in such a way that it may easy to evolve to meet the changing needs or enhancements in future.

1.8 Project Schedule

Project schedule is one of the most essential parts in a system development. Many problems will occur due to activities are not been well planning and scheduling at the beginning of the project development. Figure 1.1 shows the Gantt chart of the CLT project. Each task is divided into several phases and each phase has its own activities. Project duration: June 10 through Feb 20.

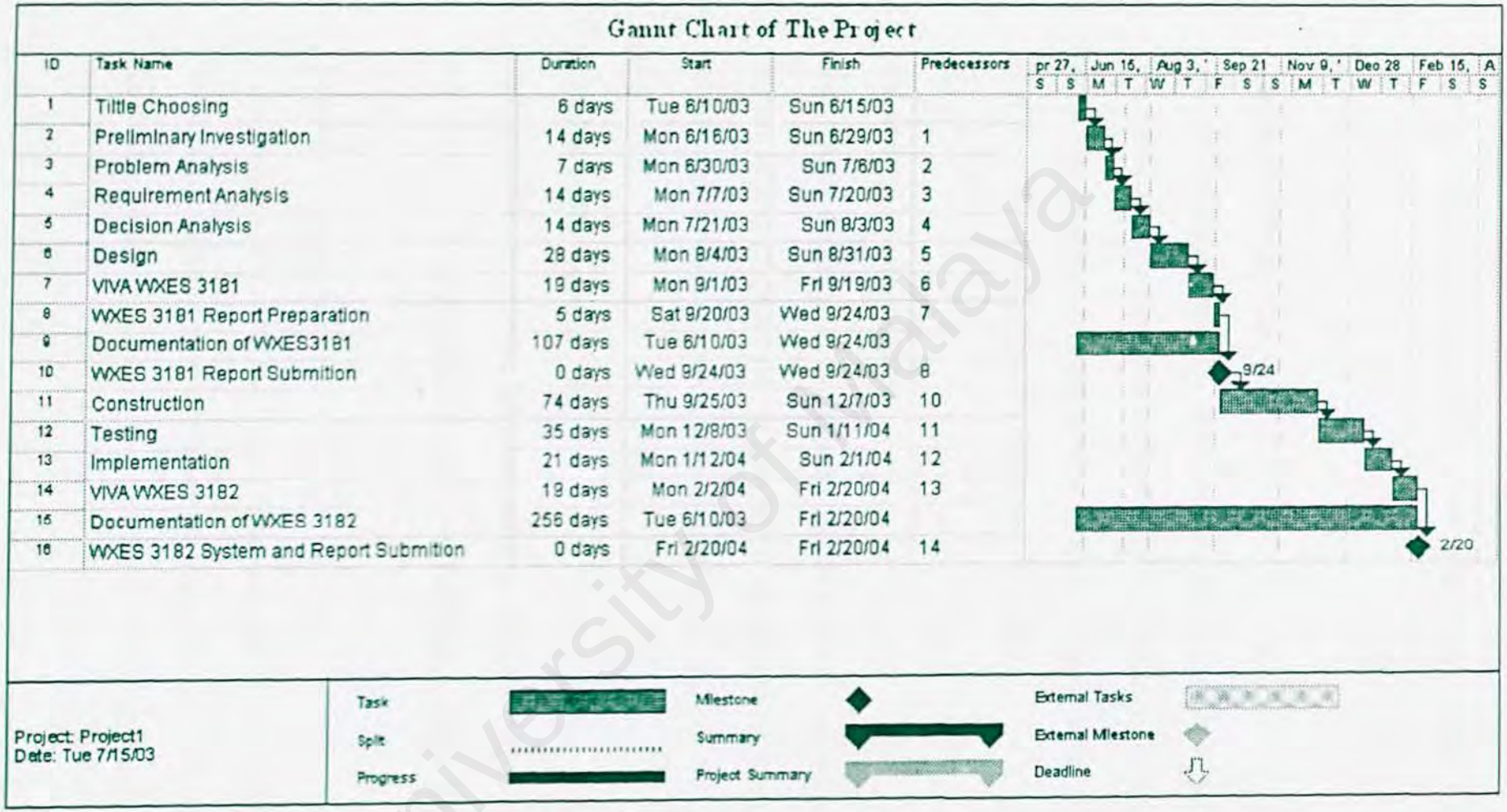


Figure 1.1: Gantt chart of CLT project

1.9 Report Layout

This report aims to describe the project intended to be carried out. The layout of the report is as depicted below: -

Chapter 1: Introduction

This chapter begins with a briefly introduction to Collaborative Learning Tool (CLT). It is followed by the motivation to develop the CLT, its problems definition, objectives and scope, chosen methodology, and its expected outcome. A project schedule is then performed in Gantt chart drawn using Microsoft Project.

Chapter 2: Literature Review

This chapter discusses the overall research that had done on this project. Firstly, the domain studies on the project are covered. It is followed by the technology review, which are potential to be used in this project development.

Chapter 3: Methodology

This chapter is started with a brief description of the software process model. Then, the overview as a whole on the justification of methodology is depicted in details.

Chapter 4: System Analysis

This chapter describes the fact-finding techniques that used to elicit the requirements. The uses cases diagram is used to define the functional requirements. It is followed by the non-functional requirements, hardware and software requirements.

Chapter 5: System Design

This chapter will include the design of system architecture, system functionality design, database design an interface design.

Chapter 6: System Implementation and Development

This chapter will describe on how to construct the application including the system development environment, program coding and database development.

Chapter 7: System Testing

This chapter describe on the techniques used to test the application.

Chapter 8: Evaluation and Conclusion

This chapter concludes the development of the application and includes some of the suggestions for future enhancement.

1.10 Chapter Summary

Collaborative learning itself is very complex and hard to implement. With the appearance of computer supported CLT, many web-based collaborative learning tools had been emerged. CLT is a tool to increase understanding of Collaborative Learning. This project is invented to stimulate two collaborative learning methods: Jigsaw Puzzle and Think-Pair-Share. Several problems of current collaborative learning tools are identified to be solved in order to achieve the expected outcome of CLT. Rational Unified Process is chosen as project methodology and the project schedule is well planned to ensure the project to be completed in time and meet those requirements.

CHAPTER 2: LITERATURE REVIEW

2.1 Domain Studies

Researches are made on switch in traditional education towards using web as a medium in education. These can help to gain more understanding on the proposed system, which will be developed.

2.1.1 Traditional Education

In the traditional classroom, especially for primary and secondary school, a teacher conveys his/her knowledge to the students at a prearranged time and location. During the class, the teacher would give a lecture according the textbook or some reference books. Simultaneously, the teacher will write down important notes or points on the blackboard or even distribute lecture notes when necessary. Exercises will be given to the students at the end of the class as revision. Sometimes, quizzes, tests or examinations are also held in the class to evaluate students' understanding for a subject. Although there are some of the teacher had implied the group work in the class but the specific technique such as collaborative learning is rarely used.

2.1.2 New Concepts of Education

With the advent of the Internet, traditional learning is in the process of undergoing a paradigm shift within the learning environment. Teaching and learning are no longer constrained to the traditional classroom. Computer-assisted learning, computer-based education, collaborative learning have brought a new approach that is having increasingly important impact on teaching or learning practices in the education.

Computer-mediated communication such as email, World Wide Web and Whiteboard environment also can help to enhance the interaction between students and teachers.

2.1.3 The Web as a Medium for Teaching and Learning

The Web has been widely touted as a means for enhancing teaching and learning *[Hawkridge 1996, p. 5; Sangster 1995, p. 3].* Indeed there are now many examples of University courses that have been launched on the Web.

2.1.3.1 Benefits

The main reasons of using the Web as a medium for education: -

- **It makes a large volume of valuable resources instantly available.**

  The Web's user-friendly hypertext graphical interface has empowered many people to use it to 'publish' information.

- **It facilitates effective communications between the teacher and student**

  Lectures could be complemented by presentation using a Web browser to access material in real-time.

- **It complements traditional teaching and learning methods**

  Discussion groups can be conducted between geographically dispersed individuals and the field experiments can be conducted on-line regardless of the time of day.

2.1.3.2 Problems

There are still some problems on using the web as a medium for education: -

- **Time to markup material is longer than the traditional lecture material**

  Longer progress times need for design issues associated with on-line material and the time required locating the most appropriate on-line resources.

- **Huge overheads associated with the maintenance of the markup material**

  Due to the rapidly changing technology associated with a subject, it was necessary to constantly evaluate the material to ensure it was relevant and up-to-date. Secondly, the volatile nature of URL links meant that regular checking was required to ensure that they were still valid.

- **Difficult to check the quality of the material**

  Hence, teachers and students might need to be able to verify the quality of information if there is any doubt as to its authenticity.

2.1.4 Collaborative Learning

2.1.4.1 Definition of Collaborative Learning

Collaborative learning is the idea that small, interdependent groups of students work together as a team to help each other learn *[Damon, 1984; Gabbert et al. 1986; Johnson and Johnson, 1989; Johnson et al. 1991; Kadell and Keehner, 1994; Kaye, 1991; Klemm, 1994; Webb, 1982].*

The concept of collaborative learning, the grouping and pairing of students for the purpose of achieving an academic goal has been widely researched and advocated throughout the education institutes. Many collaborative learning methods were explored and introduced by some researchers *[Slavin, R., Sharan, S., Lazarowitz, R. H., Webb, C., and Schmuck, R., 1985; Slavin, R. E., 1995].*

The term "collaborative learning" refers to an instruction method in which students at various performance levels work together in small groups toward a common goal. The students are responsible for one another's learning as well as their own. Thus, the success of one student helps other students to be successful.

Collaborative learning is a personal philosophy, not just a classroom technique. There is a sharing of authority and acceptance of responsibility among group members for the group's actions. The underlying premise of collaborative learning is based upon consensus building through cooperation by group members, in contrast to competition in which individuals best other group members *[Panitz 1997].*

2.1.4.2 Advantages of Collaborative Learning

Advantages of the collaborative learning can be divided into 2 aspects: -

- **Benefits Focusing on Academic aspect**
  - *Helped understanding.* Collaborative learning helps to build more positive heterogeneous relationships and encourages diversity understanding.
  - *Stimulated thinking.* Collaborative learning stimulates critical thinking and helps students clarify ideas through discussion and debate.

- *Pooled knowledge and experience.* Majority and minority populations in a class learn to work with each other had contributed positively to the learning process.
- *Got helpful feedback.* Collaborative learning medium provided students with opportunities to analyze, synthesize, and evaluate ideas cooperatively.

➢ **Benefits Focusing on Social and Emotional Aspects**

- *More relaxed and positive atmosphere makes problem-solving easy.* Collaborative learning creates an environment of active, involved, and exploratory learning.
- *Develops social communication skill.* Collaborative learning develops social interaction skill, fosters and develops interpersonal relationships and promotes positive societal responses to problems and fosters a supportive environment within which to manage conflict resolution.
- *Involve students actively in the learning process.* Collaborative learning allows students to exercise a sense of control on task.
- *Student centered instruction increase students' self esteem.* Collaborative learning encourages student's responsibility for learning.

2.1.4.3 Disadvantages of Collaborative Learning

The following are some of its disadvantages of the collaborative learning: -

➢ **Time-consuming**

Some students will feel that they wasted a lot of time explaining the material to other group members.

- **Training in collaboration learning methods is needed.**

  Few teachers or students have had any exposure to the collaboration learning teaching/learning technique. Teachers are not trained during their certification processes in collaborative methods.

- **Loss of control in the classroom**

  Major obstacle to collaboration learning is that many teachers loss control of the class if they give more responsibility to the students on their learning.

- **Large class sizes is difficult to setup with collaborative learning**

  There is a growing trend in education institutes to attempt to economize by increasing class. This runs completely contrary to the nature of collaborative learning where smaller, student centered groups have access to an instructor.

2.1.5 Methods of Collaborative Learning

2.1.5.1 Principles of Collaborative Learning Methods

Collaborative learning method is based upon the following principles:

1. Working together results in a greater understanding than would likely have occurred if one had worked independently.
2. Spoken and written interactions contribute to this increased understanding.
3. Opportunity exists to become aware, through classroom experiences, of relationships between social interactions and increased understanding.
4. Some elements increased understanding is idiosyncratic and unpredictable.
5. Participation is voluntary and must be freely entered into.

Based on some of the researches, the collaborative learning progress might follow a fundamental cycle of stages as shown in figure 2.1.

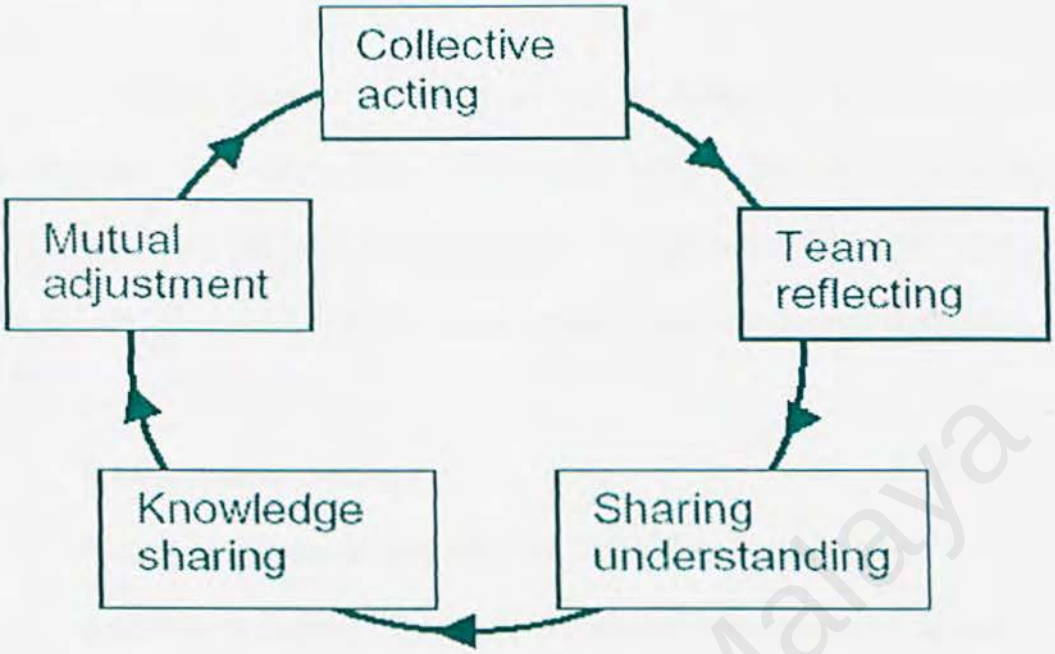


Figure 2.1: Collaborative learning cycle based on Kolb's (1984)

2.1.5.2 Examples of Collaborative Learning Methods

There are a number of collaborative learning methods used worldwide nowadays. However, this project is only focus on Think-Pair-Share and Jigsaw Puzzle. Some of other collaborative learning methods: Relate-Create-Donate, Four Corners, Multi-Voting, Nominal Group Technique, Mirroring etc.

2.1.5.3 Think-Pair-Share

Think-Pair-Share is one of the most common collaborative learning structures. It was first proposed by Frank Lyman (1981) of the University of Maryland. This is a

relatively low-risk and short collaborative learning structure, and is ideally suited for instructors and students who are new to collaborative learning.

Think-Pair-Share is one of the easiest to implement, and is adopted most quickly and widely. The structure is extremely versatile because it can be used for review and recall. The steps are simple, but it is important to follow the steps exactly to avoid the "group work" pitfalls. Below are the Think-Pair-Share's procedures: -

1. Teacher poses a question.
2. Student given time (at least 10sec) to **THINK** their own answer.
3. Each student discusses (**PAIR**) his/her answer with partner on the topic.
4. Randomly on a few students to **SHARE** their ideas with the class.

2.1.5.4 Jigsaw Puzzle

The jigsaw classroom is a specific cooperative learning technique. Just as in a jigsaw puzzle, each piece—each student's part—is essential for the completion and full understanding of the final product. If each student's part is essential, then each student is essential; and that is precisely what makes this strategy so effective. Below are the procedures of the Jigsaw Puzzle: -

1. Divide students to 5- or 6- person jigsaw groups (diverse gender, race and discuss ability).
2. Appoint 1 student (most mature) from each group as leader.
3. Divide the day's lesson to 5-6 segments.

4. Assign each student to learn a segment only.
5. Give students time to read over their segment and become familiar with it.
6. Form temporary "expert groups" by having 1 student from each jigsaw group join other students assigned to the same segment. Give time to discuss and to rehearse the presentations they will make to their jigsaw group.
7. Bring the student back to their jigsaw groups.
8. Ask each student to present her/his segment to the group. Encourage asking question for clarification.
9. Float from group to group, observing the process.
10. At the end of the session, give a quiz on the material.

2.1.6 Limitation of Traditional Collaborative Learning Approaches

Collaborative learning is a slow and tedious process, which requires participants to be present at a same place at a same time. In manual approach, it is very difficult to keep track of how group discussion is performing. It is also very hard to determine how each member is fulfilling the assigned role. Instructor will soon feel himself very hard to monitor gradation of each group's discussion concurrently.

Moreover, instructor is hard to give on the spot advises or comments to participants in respond to their misunderstanding of the topic. Besides, increasing amount of data is collected today during studies in which students and educators are engaged in learning activities using information technology and other tools. Hence, the need for tools, which can automate the progress of the collaborative learning process, is of increased importance.

2.1.7 Web-based Collaborative Learning

2.1.7.1 Introduction of Web-based Collaborative Learning

Web-based learning is assuming an increased role in education. With the rise of internet technologies and integrated office environments, web-based collaborative learning tools appear as well. The Internet allows individuals and groups to time shift their communications and extinguish physical distance, opening up new possibilities for collaborative learning. Learning in groups is available without the constraint of geographically dispersed. Participants can meet across time on the Internet, rather than in face-to-face classroom events. This can resolve the limitations of the traditional approach.

2.1.7.2 Advantages of Web-based Collaborative Learning

Some specific advantages of web-based collaborative learning are as following: -

- Save time and no traveling needed
- More immediate feedback.
- Teach and learn from anywhere.
- Provide an interactive and challenging learning environment for the learner.
- Expose learners to real-world learning experiences involving meaningful and purposeful learning.
- Inter- and intra-group collaboration took place and the system enabled this to be observed by the lecturer, who could join in discussions as required.

2.1.7.3 Limitation of Web-based Collaborative Learning

The power of Internet technologies has bringing impressive effect in collaborative learning in web-based environment. However there are still plenty of the limitations of web-based collaborative learning: -

- A major problem area has been the slow, and sometimes broken, internet connections which created frustration for some students.
- The volume of usage was much greater than was anticipated. This due to number of levels in the system considerable time needed to be spent to ensure that the teacher answered all communications in a timely manner.
- Encouraging students to use the system in an appropriate way and in a way that will enhance their learning experience was also been problematic. Early analysis of usage patterns indicated that the majority of postings elicited no replies and did not grow into threaded discussions.

2.1.8 Existing System Review

A few case studies are done on three existing collaborative learning system: Blackboard System, coMentol Workshop, and NetOp School. The purpose is to discover the most suitable approach to simulate learning environment in CLT.

2.1.8.1 Case Study 1: Blackboard System

a.) Introduction to Blackboard System

The Blackboard system is an integrated set of web-based tools designed for the creation and management of a learning environment. These tools include (this project will only focus on management as a whole): -

- Course development and management tools;
- Personal information management tools;
- Academic web resources;
- System management tools.
- Publication of learning materials (e.g. links to module related websites);
- Publication of announcements;
- Communication tools including email.

In this system, a lecturer can build up a course site with different types of learning materials. Moreover, the lecturer can use a range of communication tools to assist with the management and assessment of the module. Students can share files and use communication tools to communicate with other students and the lecturer either synchronously or asynchronously (http://www.blackboard.com/).

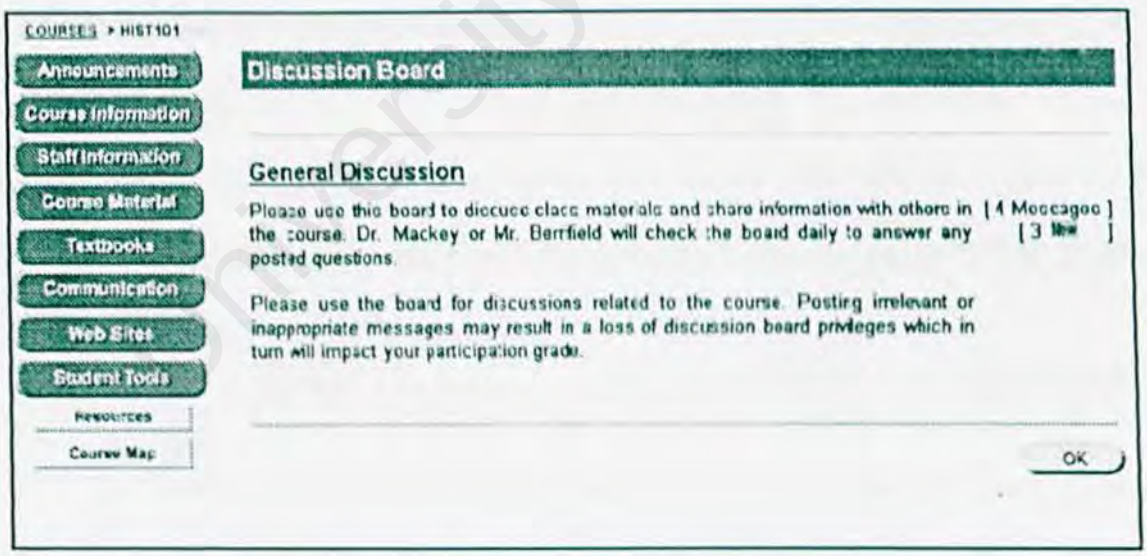


Figure 2.2: Discussion Board page of Blackboard (Access to other parts is via the buttons on the left hand side)

b.) Advantages of Blackboard System

Some specific advantages of using the Blackboard system have been identified: -

1. Management of the cohort. The onus on forming into groups, topic selection and identification of slots for tutorials and presentations has been significantly eased.
2. Communication between lecturer and student has been greatly enhanced with the use of the bulletin board.
3. Inter- and intra-group collaboration took place and the system enabled this to be observed by the lecturer, who could join in discussions as required.

c.) Problems of Blackboard System

Some of the problems of the blackboard system are similar to the limitation of the web-based collaborative learning that is stated earlier (2.1.7.3)

2.1.8.2 Case Study 2: coMentol Workshop

a.) Introduction to coMentol Workshop

coMentor is a site on the WWW to help support a course. It contains resources and information to support course and enables user to chat and help each other with coursework. coMentor can be used from any computer connected to the WWW. Figure 2.3 shows the main page of the coMentor.

For security and authentication, all user may need to login before can go into the workshop. coMentor consists of several different areas, each holding different kinds of tools and materials to support student courses.

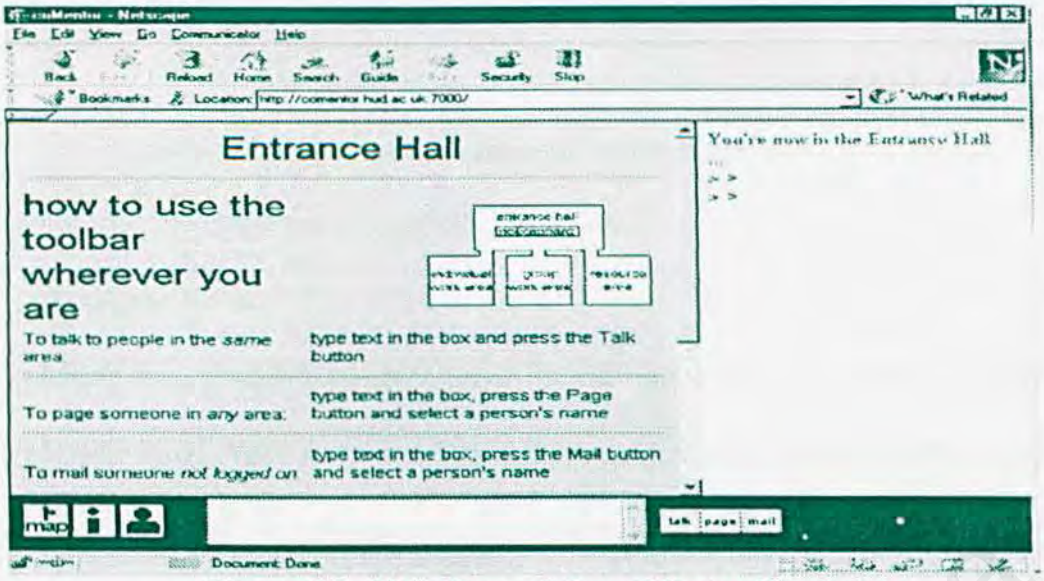


Figure 2.3: Main Page of the coMentor Workshop

coMentor provide 3 different ways to communicate with other people. First, user type in what they want to say in the text box on the toolbar and choose: -

- **"talk"** to talk privately about the contents of a particular area with everyone else who is currently looking at it.
- **"page"** to send private messages to people who are connected to coMentor, but are in a different area.
- **"mail"** to leave messages for people who are not currently connected to coMentor.

b.) Managing Users

There are 4 kinds of coMentor users: -

- **Staff user:** is exactly the same as a student user, except that she/he belongs to the category "staff", which can access to groups and individual work.
- **Student user:** the basic kind of user

- **Administrator:** there is only one Administrator.
- **Guests:** can connect to coMentor with the user name guest and no password. Fewer abilities than student/staff users.

c.) Managing Room

coMentor comes with 5 standard areas: the Entrance Hall, Noticeboard, Groupwork Area, Individual Work Area, and Resource Area. User may want to add other areas to suit their needs. If user choose to make new areas user must change the existing map of coMentor to show them. This requires user to make an image map. User can always seek help from a web page creation expert on this.

d.) Advantages of coMentor Workshop

1. The feature of the system allows user to skip from waiting until the browser finishes loading when using the WWW.
2. This is a restricted mail system. It is not possible to e-mail into coMentor from outside the system and secure the system.

e.) Disadvantages of coMentor Workshop

1. It is not easy to use as the interfaces are not so user-friendly.
2. The requirements for the installation/download for this program may need a lot of specification that limit the users to use it.

2.1.8.3 Case Study 3: NetOp School

a.) Introduction to NetOp School

NetOp School is software for Networked Classrooms. It enables a teacher to increase student time-on-task by being able to monitor their PC activity while they work. Teachers can also enhance classroom interactivity by broadcasting live demo screens to participants; mark up the featured screen to highlight a lesson; create monitored chat rooms or allow students to send private questions via an instant message button located on student PC. It provides many other useful tools to help instructors multi-task their efforts including the ability to send coursework and files to all student PCs, remote control students for one-on-one instruction, shut down or restart PCs, broadcast and control multimedia files and view classroom in several different ways.

Figure 2.4: Student Group View of NetOp School

b.) Teacher module - Superior Classroom management

- **Detail View** – view students as a list for a quick view of whose present and read private instant messages and questions from r students.

- **Mosaic View** – view all students' screens as thumbnails. Click on one or more students to start a session. Sort the thumbnails alphabetically in ascending or descending order.
- **Classroom View** – create a virtual seating plan using own classroom-lat.
- **Student Group View** – prepare and manage groups for classes in a tree-view of students. Send messages to available students and invite them to join the class.

c.) Student Module

- **Auto join class** – Automatically connect student computers to a specific class when loading the student module.
- **Browse for classes** – Lets a student look for classes listed locally or on a public server.
- **Request help** – Students can send the instructor a private request help instant message.

d.) Advantages of NetOp School

1. Increase Student Time-on-Task by having the ability to quickly and easily watch exactly what they are doing on their PCs.
2. Increase classroom interactivity by providing demos or instruction to one, several or all students at once; students can send a private message/question.
3. Easily Multi-Task r efforts by doing much more with r students, computers and coursework - all from a desk.
4. Easy to install and use, and less expensive.

e.) Problems of NetOp School

1. Internet connections sometime are slow, and sometimes broken
2. Help provided to use the system is insufficient.
3. Difficult to monitor students when group is large.

2.1.9 Proposed System

From the case studies that are revised, a few common limitations are found in the existing system. These include: -

- These systems do not apply Collaborative Learning technique fully.
- It is difficult to operate the system for novices.
- The system is not cost effective for secondary school.

As a conclusion, CLT is needed and will be developed to resolve all of those limitations. Besides, it will try to integrate some of the advance functions to make it more powerful tool.

2.2 Technology Review

CLT is a web-based system which involves Client and Server computing. The following sections will discuss on the technology reviews, which are considered related the CLT development.

2.2.1 Internet

2.2.1.1 Introduction to Internet

Internet is a worldwide collection of linked computer systems that can be accessed by computer users with a connection to the network. It is originates from a project undertaken in the 1960s by the United States Defense Department, which explored ways of protecting its computer network from nuclear attack. The idea was to ensure that, if any single computer system in its large network was destroyed, data could be relayed to the other systems via different paths.

As the processing power and availability of personal computers constantly increases, the Internet continues to grow exponentially in terms of the amount of stored data available, the volume of traffic and with respect to the number of users. The globalization of the world economy has meant increased demands for fast and reliable data transfer causing technological advances in data communications to be quickly integrated. These demands will spur the development of the Internet and will continue to push its expansion and development throughout this present century.

2.2.1.2 The Main Uses of the Internet

- **Access to Information:** The Internet allows users to access information on any subject imaginable. The search engines ease the searching process.
- **International Communication:** The Internet allows users to communicate exchange and share information through network using Email, Fax, Netmeeting, Teleconferencing, Chat and Instant Messaging.
- **Education:** Besides learning new things and gaining knowledge, users will become more proficient at navigating computer tools.
- **Fun:** Fun and games abound in the Internet.

- **Profit / Business & Commerce:** The Internet can be an effective medium for business communications.

2.2.1.3 World Wide Web (WWW)

As the information stored on computers linked to the Internet began to increase in volume and variety of formats, issues of accessibility and ease of use arose. In 1990, Tim Berners-Lee of CERN (the European Laboratory for Particle Physics) placed three programs (HTML, HTTP and URL) on the Internet, successfully developed the WWW and several communication protocols that form the backbone of the Web. This allows previously widely incompatible data to be more readily accessible.

The WWW is effectively any data located on a server computer that is accessible to the general population through a browser. Increasingly the earlier reliance on HTML as a foundation for all Web page implementation is shifting in favors of protocols that support database and multimedia functionality. Newer languages such as XML are now improving the way the Web is delivered to end users by increasing download speeds and broadening the dynamic characteristics of highly interactive usage.

Although the WWW is central in the common use of the Internet, it is often mistaken by many as being synonymous with it where as in fact as detailed above the WWW is an example of one of the many ways the Internet can be utilized. Understanding the distinction between the two is important in gaining a fuller understanding of the online world.

2.2.2 Web Server

A Web server is a program that using the client/server model and the World Wide Web's Hypertext Transfer Protocol (HTTP), serves the files that form Web pages to web user that forward their request. Every computer on the Internet that contains a web site must have a Web server program (or else the site files must be sent to a computer that has a Web server program).

Web server often come as part of a larger package of Internet- and Intranet-related programs for serving email, downloading request for FTP files, and building and publishing Web pages. Consideration in choosing a Web server include how well it works with the OS and other servers, its ability to handle server-side programming, publishing, search engine, and site building tools that come with it.

2.2.2.1 Internet Information Server (IIS)

Microsoft's Internet Information Server is an Internet file and application server (enterprise-level Web server). It is actually a group of Internet server (HTTP, FTP, SMTP, and NNTP). IIS can be used alone as a Web server, to in conjunction with compatible technologies to set up Internet commerce, to access and manipulate data from a variety of data sources, and to build web applications that take advantage of server script and component code to deliver client-server functionality. With IIS, Microsoft includes a set of programs for building and administering Web sites, a search engine, and a support for writing Web-based applications that access database.

IIS includes security features and promises that it is easy to install. It works closely with the Microsoft Transaction Server to access databases and provide control at the transaction level. IIS is the most popular commercial Web server for Windows 2000 in use today. While posting a Web page to IIS is easy, configuring it to ensure maximum performance and security for r applications can be challenging.

2.2.2.2 Microsoft Personal Web Server (PWS)

Microsoft's Personal Web Server (PWS) is a scaled-down version of the commercial Information Internet Server (IIS) included with the Server edition of Microsoft Windows NT. PWS is a great entry-level Web server that makes it easy to publish personal home pages, serve small Web sites, and share documents via a local intranet. One key advantage to using PWS over IIS and similar high-end Web servers is the client's ease of use. PWS is one of the best servers available for providing help wizard and running quickly.

PWS does include support for Active Server Pages (ASP), script debugging, and etc important features found in its commercial sibling. Additionally, PWS presents the ability to develop transactional Web applications using the Microsoft Transaction Server. Overall, while most large enterprises will likely bypass Microsoft's PWS for the high-end IIS; PWS will remain one of best available options for individuals wanting to serve their own personal home pages and for small organizations needing to host their own Web sites.

2.2.2.3 Apache Web Server

The Apache Web Server, maintained by the Apache Software Foundation, is currently the most popular Web server because of its stability, efficiency and portability. Additionally, the Apache server is robust, fast and become the true spirit of the Internet. It is an open source product (i.e., software that can be freely obtained and customized) that runs on UNIX, Linux and Windows platforms.

Apache Group has recently been focusing on Apache 2.0. There have been many changes to the Apache code since Version 1.3. Some of these changes make an administrator's job easier and some make it harder; however, the changes are all designed to make Apache the most flexible and portable Web server available.

2.2.3 Web Browser

A Web browser is a program to view and explores all the information on the Web. The first web browser with a graphical user interface was invented by Mosaic, in 1992. Due to demand for more advanced usage, browsers that could interpret images, moving graphics and sound files have been produced and are repeatedly upgraded. The most generally used at present are Microsoft Internet Explorer and Netscape Navigator. These two browsers are highly competitive and the only browsers that the vast majority of Internet users are aware of.

2.2.3.1 Microsoft Internet Explorer

Internet Explorer makes it easier to get the most from the World Wide Web, whether to search for new information or browsing the favorites Web Sites. Microsoft Internet

Explorer built-in IntelliSense technology can save the time completing routine web tasks, such as automatically completing web address and forms for, and automatically detecting network and connection status. It is currently the most popular Web browser. It comes with the Windows operating systems and the latest version of the Macintosh operating system, Mac OS X.

2.2.3.2 Netscape Navigator

Netscape is the name of a company, Netscape Communication, now owned by America Online (AOL). Netscape was initially prime product in terms of usability and number of users. It was developed in 1995 by a team led by Marc Andersen, who created Mosaic, at the University of Illinois' National Center for Supercomputing Application (NCSA) in 1993.

A primary source of revenue for Netscape and AOL is the Netscape line of Web Server products that it develops and has marketed on the success of its wide-scale browser usage. Besides, Netscape and AOL also envision the Netscape Web Site, now transformed into a leading Web Portal, as a leading source of revenue through adverting and e-commerce.

2.2.4 System Architectures

2.2.4.1 Client/Server System

As a matter of fact a client server system is a special case of a co-operative computer system. All such systems are characterized by the use of multiple processes that work together to form the system solution. There are two types of co-operative systems:

client-server systems and peer-to-peer systems. The client and server systems consist of three major components: a server with relational database, a client with user interface and a network hardware connection in between. Client and server is an open system with number of advantages such as interoperability, scalability, adaptability, affordability, data integrity, accessibility, performance and security.

a.) What Do the Client Programs Do?

The client-based process is the application that the user interacts with. It contains solution-specific logic and provides the interface between the user and the rest of the application system. In this sense the graphical user interface (GUI) is one characteristic of client system. They usually deal with:

- managing the application's user-interface part
- confirming the data given by the user
- sending out the requests to server programs
- managing local resources, like monitor, keyboard and peripherals

b.) What Do the Server Do?

Its purpose is fulfilling client's requests. What they do in general is:

- receive request
- execute database retrieval and updates
- manage data integrity
- sent the results to client back
- act as a software engine that manage shared resources like databases, printers, communication links

c.) Examples of Client and server systems

The most common client and server systems are: finger, telnet, ftp, NFS (Network File System), X Window System, Gopher, and World Wide Web.

2.2.4.2 Architectures for Client/Server System

Both traditional Client/Server as well as net-centric computing is tiered architectures. In both cases, there is a distribution of presentation services, application code, and data across clients and servers. There is a networking protocol that is used for communication between clients and servers. They support a style of computing where processes on different machines communicate using messages. In this style, the "client" delegates functions or other tasks (such as data manipulation logic) to one or more server processes. Server processes respond to messages from clients. A Client/Server system has several layers, which can be visualized in either a conceptual or a physical manner. Viewed conceptually, the layers are presentation, process, and database. Viewed physically, the layers are server, client, middleware, and network.

2.2.4.2.1 Client/Server 2-tiered architecture

2-tiered architecture is also known as the client-centric model. Nearly all of the processing happens on the client, and client accesses the database directly rather than through any middleware. Here the client talks directly with the server; there is no intervening server between. It is quite good for small environments with not so many users. It is also known as first generation architecture. 2-tiered architecture has low development and low purchase costs. It is also used in prototyping, but this is a bit risky, since real life can bring unexpected surprises in implementations.

2-tiered architecture is the simplest one to implement. It is the most stable form of Client/Server implementation. Direct access to the database makes it simpler to verify the test results. However this model is limited by the scalability and difficulties for maintenance. Because it doesn't partition the application logic very well, changes require reinstallation of the software on all of the client desktops.

2.2.4.2.2 Modified 2-tiered architecture

Because of the nightmare of maintenance of the 2-tiered Client/Server architecture, the logic is moved to the database side, implemented using triggers and procedures. This kind of model is known as modified 2-tiered architecture. Modified 2-tiered architecture is more complex than 2-tiered architecture. Another complication is dynamic database queries. They are constructed by the application and exist only when the program needs them.

2.2.4.2.3 Client/Server 3-tiered architecture

For 3-tiered architecture, there is a server, or third company between client and server. It may provide translation, metering and/or intelligent agent services. The application is divided into a presentation tier, a middle tier, and a data tier. The middle tier is composed of one or more application servers distributed across one or more physical machines. The characteristics make the 3-tiered architecture desirable as a development and implementation framework. It is more scalable than 2-tiered architecture; it provides separate presentation and logic process, as well as multi-process and multi platform. It is also known as second generation architecture.

3-tiered architecture may have additional layer of application servers. This let the application logic split on different servers, with one specialized server for process handling. This decreases the load on the system, since the client doesn't need to contact every server by sending individual requests; instead it sends one whole request that covers everything. However the trade-off is 3-tiered architecture is hard to learn, compared to 2-tiered, there is a language barrier, it increases the complexity of system.

2.2.5 Web Application Development Tools

2.2.5.1 Microsoft FrontPage

Microsoft FrontPage is WWW authoring and management tool that requires no programming knowledge but is robust enough for experienced web site developers. It includes the FrontPage Explorer, which allow users to view and manage the web site, and the FrontPage Editor for creating and editing web pages without needing to know HTML.

In FrontPage, users can use the graphical FrontPage Explorer to create, view, and maintain their FrontPage webs and to publish them on the computer, Local Area Network (LAN), or the Internet. The FrontPage Explorer has commands for administering FrontPage webs, testing and repairing hyperlinks, viewing all of a FrontPage web's files and folders, importing and exporting files, and launching the FrontPage Editor and other applications to create and edit FrontPage web's contents.

2.2.6 Programming Language

2.2.6.1 Markup Language

Hypertext Markup Language (HTML) is the *lingua franca* for publishing hypertext on the World Wide Web. It is a non-proprietary format, based upon SGML, for describing the structure of hypermedia documents - plain text (ASCII) files with embedded codes for logical markup, using tags like <A> and </A> to structure text into tables, hypertext links interactive forms, headings, paragraphs, lists, and more. It can be created and processed in a wide range of tools from simple plain text editors to sophisticated WYSIWYG authoring tools.

Extensible Markup Language (XML) is a human-readable, machine-understandable, general syntax for describing hierarchical data, applicable to a wide range of applications (databases, e-commerce, Java, web development, searching, etc.). Custom tags enable the definition, transmission, validation, and interpretation of data between applications and between organizations.

Extensible Hypertext Markup Language (XHTML) is a reformulation of HTML 4.0 as an application of XML 1.0. XHTML 1.0 specifies three XML namespaces, corresponding to the three HTML 4.0 DTDs: Strict, Transitional, and Frameset. Each of these three namespaces is identified by its own URI. XHTML 1.0 is the basis for a family of future document types that extend and subset HTML.

XHTML 1.0 is the first step toward a modular and extensible web based on XML. It provides the bridge for web designers to enter the web of the future, while still being able to maintain compatibility with today's HTML 4 browsers.

2.2.6.2 Java Script

JavaScript is a scripting language developed by Netscape to enable web authors to design interactive sites. JavaScript is different from Java. Although it shares many of the features and structures of the full Java language, it was developed independently. JavaScript can interact with HTML source code to enable web authors to spice up their sites with dynamic content. It is supported by recent browsers from Netscape and Microsoft, though Internet Explorer supports only a subset, which calls Jscript.

Using JavaScript, even less-experienced developers will be able to direct responses from a variety of events, objects, and actions. It provides anyone who can compose HTML with the ability to change images and play different sounds in response to specified events.

2.2.6.3 VB Script

VBScript is a subset of the VB Programming language. It talks to host applications using Windows Script. With Windows Script, browsers and other host applications do not require special integration code for each scripting component. Windows Script enables a host to compile scripts, obtain and call entry points, and manage the namespace available to the developer. With Windows Script, language vendors can create standard language run times for scripting.

VBScript is integrated with WWW browsers. VBScript and Windows Script can also be used as a general scripting language in other applications. It is much easier to use and learn than JavaScript.

2.2.6.4 ASP

ASP is a server extension of the IIS web server released by Microsoft. ASP allows developers to code custom tags in JScript or VBScript. These tags can be interpreted by IIS before the pages are sent out.

An ASP page at its core is simply a text file that has been named using the extension .asp and which contains HTML and scripting. Scripting, usually in VBScript provides a means to embed programmatic logic into HTML files that will be dynamically interpreted as the HTML page goes through the web server and also provides access to any server side object.

One of the most important features of ASP is that it allows user to easily access data and put it on a Web page. User can simply display data from an ODBC-compliant database, or use ASP to make decisions about what to display on a Web page. Another important ASP feature is the ability to use cookies to store and retrieve information. The Request object has a Cookie collection, and user can use this in data processing.

2.2.6.5 ASP.net

ASP .NET is the latest release in Microsoft's line of Active Server Pages. It is a set of technologies in the Microsoft .NET Framework for building Web applications and XML Web Services. ASP.NET pages execute on the server and generate markup such as HTML, WML or XML that is sent to a desktop or mobile browser. ASP.NET pages use a compiled, event-driven programming model that improves performance and

enables the separation of application logic and user interface. ASP.net combines unprecedented developer productivity with performance, reliability, and deployment.

a.) Developer Productivity

ASP.NET helps to deliver real world Web applications in record time.

- Easy Programming Model.
- Flexible Language Options
- Great Tool Support.
- Rich Class Framework.

b.) Improved Performance and Scalability

ASP.NET allow to use serve more users with the same hardware.

- Compiled execution - automatically detects any changes, dynamically compile the files if needed, and store the compiled results to reuse.
- Rich output caching - dramatically improve the performance and scalability.
- Web-Farm Session State - share session data user-specific state values across all machines in Web farm.

c.) Enhanced Reliability

ASP.NET ensures the application is always available to the users.

- Memory Leak, DeadLock and Crash Protection - automatically detect and recover from errors to ensure the application is always available.

d.) Easy Deployment

ASP.NET takes the pain out of deploying server applications.

- "No touch" application deployment
- Dynamic update of running application
- Easy Migration Path

e.) New Application Models

ASP.NET extends the application's reach to new customers and partners.

- XML Web Services
- Mobile Web Device Support.

2.2.6.6 Comparison between ASP & ASP.net

Table 2.1 Differences between ASP 3.0 and ASP .NET

ASP 3.0 (/ ASP Classic)	**ASP .NET** (/ ASP+)
Supports VBScript and JavaScript	Supports Visual Basic, C#, and JScript
Uses conventional HTML forms for data collection	Uses web forms that look like HTML forms to the client, but add much functionality due to server-side coding
Files end with *.asp extension	Files end with *.aspx extension
5 objects: Request, Response, Server, Application, Session	.NET contains over 3400 classes
Queried databases return recordsets	XML-friendly data sets are used instead of recordsets.

In additionally, ASP .NET controls can be bounded to a data source, including XML recordsets. ASP.NET is also much faster than classic ASP, while preserving the "just hit save" update model of ASP.

2.2.6.7 C#

C# is the key language for Microsoft's next generation of Windows services, the .NET platform. This new programming language is fast and modern and was designed to increase programmer productivity. It enables programmers to quickly build a wide range of applications for the new .NET platform. The .Net platform enables developers to build its components to become Web services available across the Internet.

C# combines the power of C/C++ and the productivity of Visual Basic. It is a simple, object-oriented, and type-safe programming language. Initial language specifications also reveal obvious similarities to Java, including syntax, strong web integration and automatic memory management.

2.2.6.8 VB.net

VB.NET (Visual Basic.NET) is a programming language and development tool that is part of the Microsoft .NET framework and a member of the Microsoft Visual Studio .NET family of products. With VB.NET, powerful Windows and Internet applications can be created using a unified toolkit.

Powerful new features give VB.NET the simplicity of its predecessors, but the power of creating full-fledged applications that take advantage of the computing field's latest innovations. Below are some of the VB.NET features: -

- Ease to use
- Automatic Memory Management
- Fully Object-Oriented
- Powerful Database Support
- Visual Development
- True Web Development

2.2.6.9 Comparison between C# & VB.net

Both languages (C# & VB.net) are very similar and can switch fairly easily between one and the other no matter which learn. Microsoft is marketing C# heavily because it's got a good name and has the right kinds of buzzwords to make it sound like a 'serious' programming language. The language purists prefer C# (curly braces are more 'hard-core'), but it appears that Visual Studio.NET has better VB support. However the design of .NET actually means that VB.net and C# are not very different.

In general C# does not translate any more directly into MSIL (Virtual Machine Code) than VB.net so it won't perform any faster. In addition, the same engine (no matter what programming language is chosen) does the final compilation of MSIL into raw machine instructions. Hence there is not a major difference in speed between VB.net and C#.

There are some things C# can do but not in the VB.net. For example C# can write self-documenting code but VB.net not provides such function. However these differences are relatively small.

2.2.6.10 Java

Java is a full-featured computer language that incorporates object-oriented programming. It is simpler and more robust than other computer languages. It combines features that make it ideal for programs which must deal with networks. Java is a write once/run anywhere programming language. Some advantages of Java: -

a.) Distributed

Java supports applications on networks. Java's socket class also supports reliable stream network connections, which allow creating distributed clients and servers.

b.) Portable

Java's portability comes from the neutral construction of its architecture. The Java environment can also be ported to new operating systems and hardware platforms.

c.) Multithreaded

Multitasking occurs when an operating system runs more than one program simultaneously. Java can prioritize its threads.

d.) Dynamic

Java can adapt to an evolving environment. Java programs allow for new instance variables and methods in a library's objects without affect the dependent client objects.

e.) Secure

There are three features in Java's runtime environment that make it secure: Runtime memory lat, Bytecode verifier and File access restrictions.

Although there are many advantages of Java, but most Java applications can't really challenge the speed of C because the code is interpreted, not compiled, and there are many runtime checking operations.

2.2.7 Database Management System (DBMS)

2.2.7.1 Introduction of the Database Management System

A Database Management System (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database. It is the hardware/software package that facilitates the creation and maintenance of a computerized database. It shares the data in the database among multiple applications or users. It serves as the intermediary between the user and the database by translating user requests into the complex code required to fulfill those requests. Figure 2.5 below illustrates the concept that the DBMS stands between the database and the user(s).

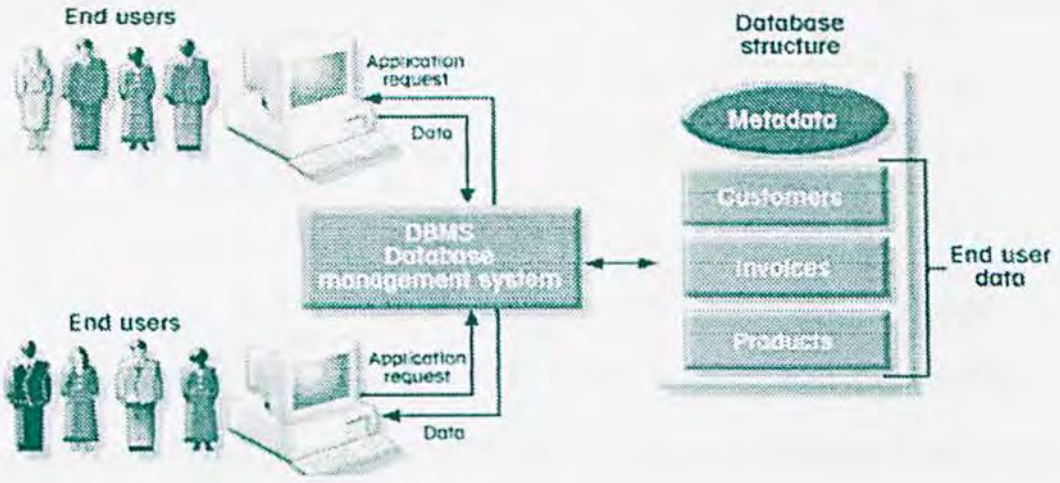


Figure 2.5: The interaction between the end user and the database

2.2.7.2 Benefits of DBMS

There are many good reasons why DBMS are important in the information-based society. The DBMS: -

- Helps manages very large amounts of data more efficient and effective.
- Contains a query language that makes it possible to produce quick answer to ad hoc queries.
- Reduce the probability of data inconsistency.
- Supports secure and atomic access to very large amounts of data.
- Controls redundancy in data storage
- Enforces integrity constraints
- Provides data abstraction through schemas.

2.2.7.3 Microsoft Access

Microsoft Access is a popular data management application that store information in tables, which it manages directly from the local disk. It is a Windows-based database management system. It is one of the programs in the Microsoft Office suite.

With Access, data can be entered, stored and manipulated in a variety of ways. It also allows querying a database for information. It can be used as a front end, that is, as an interface to information that is located elsewhere and handled by another storage management system. In this case, Access acts as a client that connects to a server that provides the data.

Access has its strengths, such as an easy to use interface. However Access also has its limitations. It is generally used as a personal or single-user application, typically for managing limited amounts of data. As its storage management limitations, it is not commonly used for databases hundreds of megabytes in size.

2.2.7.4 MySQL

The MySQL database server is the world's most popular open source database. It is a true multi-user, multi-threaded RDBMS (Relational Database Management System) server that uses SQL to interact with and manipulate data. MySQL is a client/server implementation that consists of a server daemon mysqld and many different client programs or libraries.

Its architecture makes it extremely fast and easy to customize. Extensive reuse of code within the software and a minimalistic approach to producing functionally-rich

features has resulted in a database management system unmatched in speed, compactness, stability and ease of deployment. The unique separation of the core server from the storage engine makes it possible to run with strict transaction control or with ultra-fast transactionless disk access.

MySQL server provides a few important features including multithreading capabilities that enable the database to perform multiple tasks concurrently, allowing the server to process client requests efficiently; support for various programming languages and handle large databases. It is available for Windows, Linux and UNIX.

MySQL's rising popularity benefits from the open source software movement. The term open source refers to software that can be freely obtained and customized to fulfill corporate, educational or personal requirements.

2.2.7.5 PostgreSQL

PostgreSQL is a relational database management system that has been around since the mid-eighties. Until recently it was not as feature rich or as fast as other databases such as Oracle or MySQL.

With the release of version 7.0.3, postgreSQL has leaped over MySQL, and ORACLE in many benchmarks. It is now has foreign key support (a feature that ORACLE has, but MySQL does not have). It is free and can be used for website and internal database needs. Some other advantages of the PostgreSQL including speed about the same factor as commercial databases, supports a broader subset of SQL than

MySQL, supports transactions and large tables that exceed Linux' file limit, and fully programmable. However, there are still some limitations. It is slower than MySQL, does not support the entire ANSI SQL 92' standard and it isn't hierarchical.

2.2.7.6 Oracle

Oracle is a multi-user database. It provides unprecedented ease-of-user and is pre-tuned and pre-configured for dynamic workgroup and line-of-bus environment. It includes a fully integrated set of easy-to-use management tools, full distribution, and replication and web features. Oracle also provides the highest levels of availability through fast fail over, easier management, and zero data loss disaster protection, with Data Guard, the only complete data protection solution available on the market. Oracle can runs on UNIX, Linux and Windows platform. However, it is expensive and separate licenses are required for each of its database engine.

2.2.7.7 Microsoft SQL Server

Microsoft SQL Server 7.0 is a single process, multithreaded relational database server primarily intent for transactional processing. It is based on the client/server architecture, which divides processing into two components: a front-end, or client component, that run on a local workstation and a back-end, or server component, which runs on a remote computer.

2.3 Chapter Summary

From the studies on the first part of the literature review: Domain Studies; the fact of current systems on Collaborative Learning (either in manual approach or web-based

approach) are still insufficient and inefficient to support today's educational world. This is especially for primary and secondary schools in Malaysia. Some of the problem domains for CLT have been identified. These have given some ideas on how to enhance the quality of the new system development: CLT. The second part of the literature review: technology review; give an ideas and suggestions on choosing the appropriate development tools for this project. The chosen development tools will be described in chapter 4. ASP.NET will be used to develop the CLT as its advance features exceed the ASP functions. For the system architecture, three-tiered Client-Server architecture had been defined as for CLT's architecture.

CHAPTER 3: METHODOLOGY

3.1 Software Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) is the overall process of developing information systems through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance. There are many different models and methodologies, but each generally consists of a series of defined steps or stages. Figure 3.1 illustrates stages in SDLC.

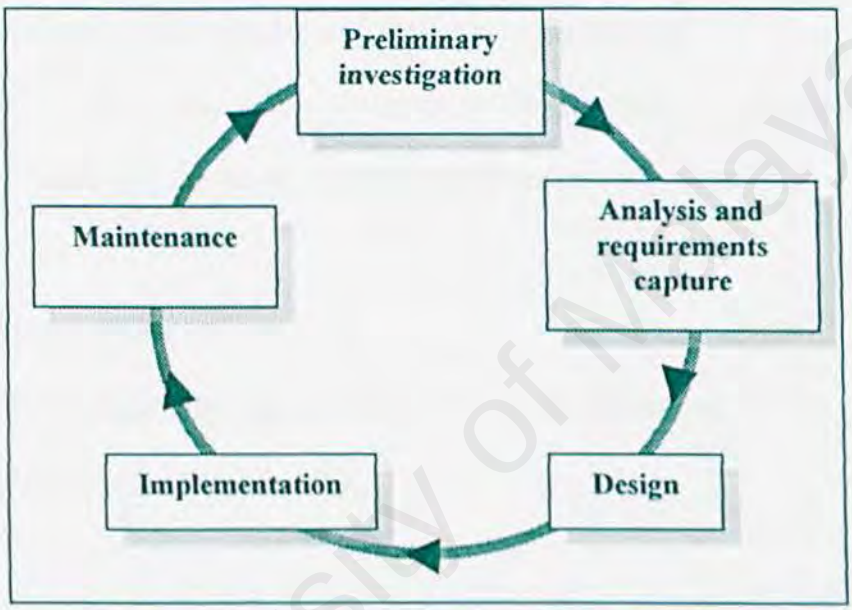


Figure 3.1 System Development Life Cycles

3.2 Software Process Model

There are many types of software process model nowadays, such as Waterfall, V-Model, Prototyping, Spiral Model, Incremental Model, Rational Unified Process and etc. Each of these models has their own benefits and drawbacks. Nevertheless, there are fundamental activities which are common to all software process.

Yet, every project has the same goals: to minimize risk, ensure expected outcome, and deliver high-quality software on time. Thus, choosing a suitable software process model becomes important. Reason for modeling a development process: -

- Written description of the development process provides a general understanding of activities, resources and constraints for a project.
- Process model helps to keep track of the development process and detect inconsistencies, redundancies and omission in the process.
- Process model should reflect the goals of development. Developer is able to evaluate activities for the appropriateness in addressing the goals.

3.3 Justification of Methodology

The methodology chosen for this project (CLT) is Rational Unified Process. It is described in details in the following part.

3.3.1 Rational Unified Process (RUP)

Rational Unified Process is a software engineering process, which provides a disciplined approach to assign tasks and responsibilities. Its goal is to ensure the design of high-quality software that meets the end-users requirements, within a schedule and budget.

RUP is an object-oriented and Web-enabled program development methodology. It is a software development process platform that delivers proven best practices and a configurable architecture. It contains a Development Kit, providing

support for configuring the process to suit the needs of an organization. It is also supported by tools, which automate large parts of the process. These tools are used to create and maintain the various artifacts of the software engineering process. Figure 3.2 shows how the emphasis of the RUP varies over time.

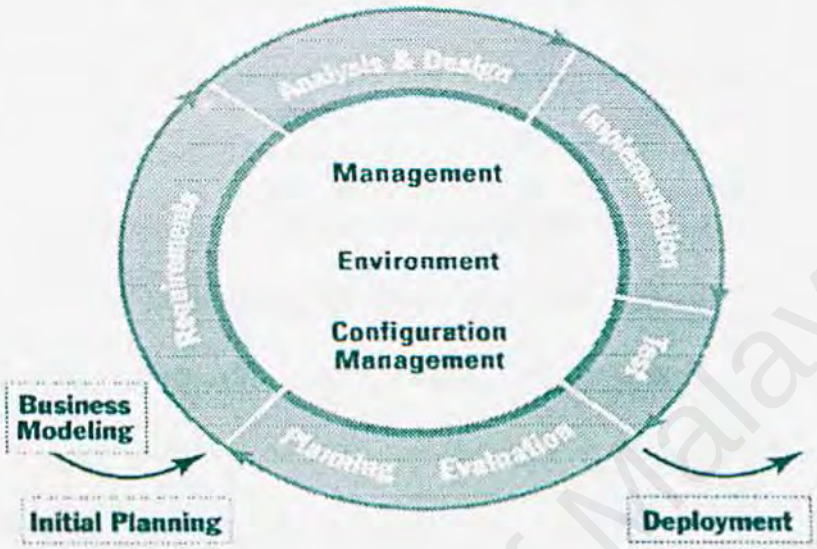


Figure 3.2: Emphasis varies over time

RUP is designed and documented using the Unified Modeling Language (UML). An underlying object model, the Unified Software Process Model (USPM) provides a very coherent backbone to the process. As it is modular and in electronic form, it can be adapted and configured to suit the specific needs of a development organization.

3.3.2 Architecture of Rational Unified Process

The RUP project structure is shown in two dimensions: -

- *Horizontal dimension*: time and lifecycle aspects of the process as it unfold.
- *Vertical dimension*: core process disciplines or workflows.

The first dimension (horizontal) represents the dynamic aspect of the process expressed in terms of cycles, phases, iterations, and milestones. In the RUP, a software product is designed and built in a succession of incremental iterations. This allows testing and validation of design ideas, as well as risk mitigation, to occur earlier in the lifecycle. The second dimension (vertical) represents the static aspect of the process described in terms of process components: activities, disciplines, artifacts, and roles.

RUP establishes four phases of development: Inception, Elaboration, Construction and Transition. Each of these phases is organized into a number of separate iterations, which must satisfy defined criteria before the next phase is undertaken. Besides, RUP also provides a prototype at the completion of each of the iteration. Figure 3.3 shows the overall architecture of the RUP.

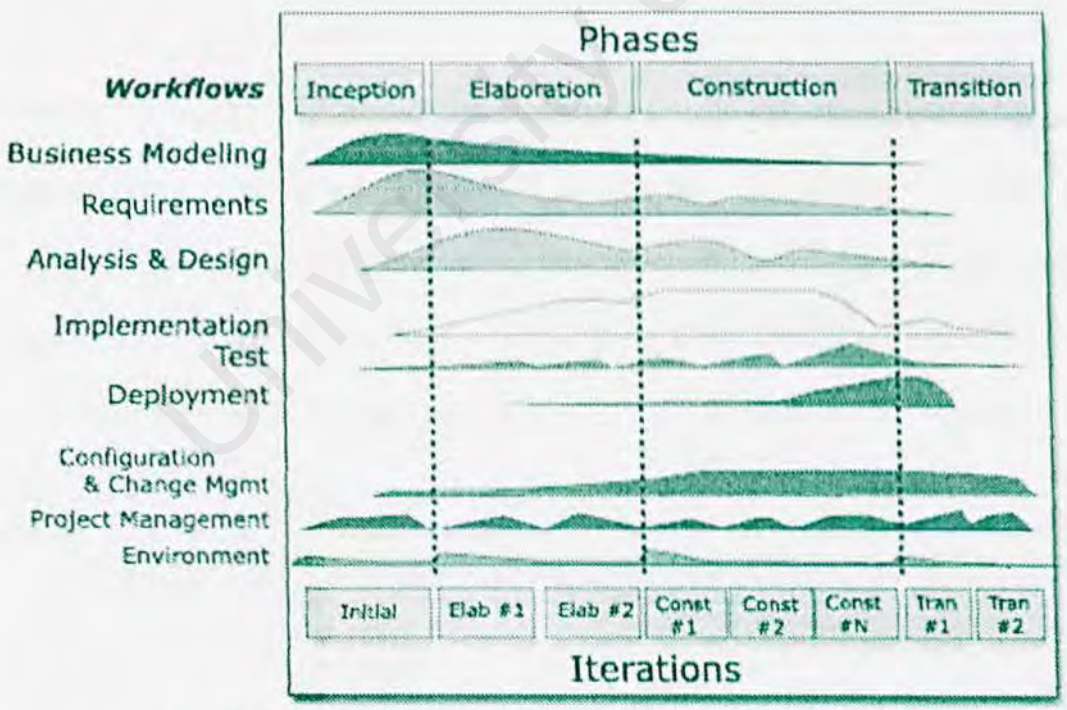


Figure 3.3: Two Dimensions of the Rational Unified Process

3.3.2.1 Phases - The Time Dimension

The software lifecycle is broken into cycles, each cycle working on a new generation of the product. The RUP divides one development cycle in four consecutive phases: Inception, Elaboration, Construction and Transition; as shown in figure 3.4 below.

Each phase is concluded with a well-defined milestone—a point in time at which certain critical decisions must be made and therefore key goals must have been achieved. Each phase has a specific purpose.

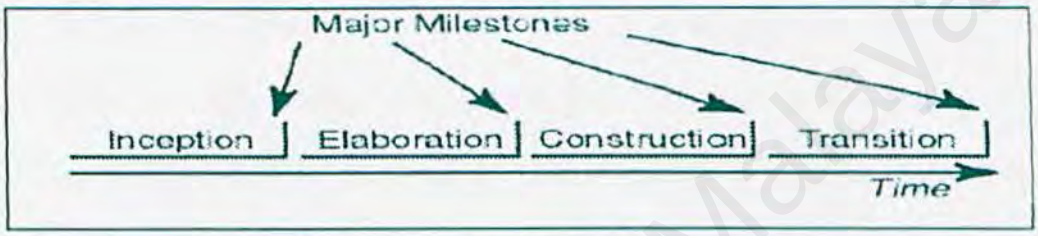


Figure 3.4: The phases and major milestones in the process.

3.3.2.1.1 Inception Phase

During the inception phase, the business case for the system is established and the project scope is delimited. All external entities with which the system will interact (actors) are identified and the nature of this interaction is defined at a high-level. This involves identifying all use cases and describing a few significant ones. The business case includes success criteria, risk assessment, and estimate of the resources needed, and a phase plan showing dates of major milestones.

Figure 3.5: Milestone of Inception phase: Lifecycle Objective

The milestone of inception phase: Lifecycle Objectives Milestone. The project may be cancelled or considerably re-thought if it fails to pass this milestone.

3.3.2.1.2 Elaboration Phase

The purpose of the elaboration phase is to analyze the problem domain, establish a sound architectural foundation, develop the project plan, and eliminate the highest risk elements of the project. This phase is considered as the most critical phase. Regularly, two prototypes are executed: -

- **Architecture prototype** – built in one or more iterations, depends on the scope, size, risk, and novelty of the project. This address the critical use cases identified in the inception phase, which typically expose the major technical risks of the project.
- **Evolutionary prototype** – does not exclude the development of one or more exploratory, throwaway prototypes to mitigate specific risks. A production-quality component is always the goal.

Figure 3.6: Milestone of Elaboration phase: Lifecycle Architecture

The milestone of elaboration phase: Lifecycle Architecture Milestone. The detailed system objectives and scope, the choice of architecture, and the resolution of

the major risks should be examine at this point. The project may be aborted or considerably re-thought if it fails to pass this milestone.

3.3.2.1.3 Construction Phase

- During the construction phase, all remaining components and application features are developed and integrated into the product, and all features are thoroughly tested. It is a manufacturing process where emphasis is placed on managing resources and controlling operations to optimize costs, schedules, and quality. The result of this phase is a product ready to use by its end-users.

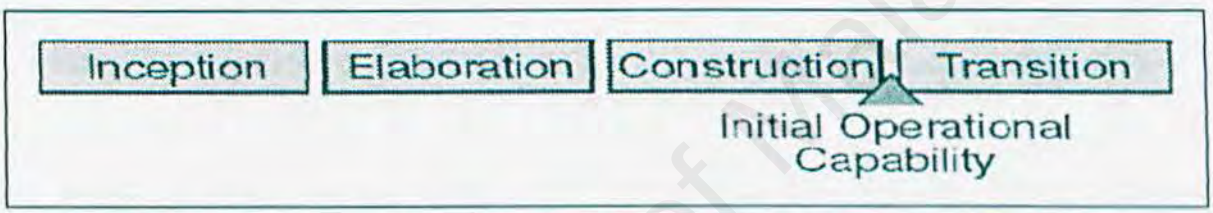


Figure 3.7: Milestone of Construction phase: Initial Operational Capability

The milestone of construction phase: Initial Operational Capability Milestone. Developer will decides if the software and the users are ready to go operational, without exposing the project to high risks. Transition may have to be postponed by one release if the project fails to reach this milestone.

3.3.2.1.4 Transition Phase

The purpose of the transition phase is to deliver the software product to the user community. This phase is entered when a baseline is mature enough to be deployed in the end-user domain.

The transition phase focuses on the activities required to place the software into the hands of the users. Typically, this phase includes several iterations. Considerable effort is expended in developing user-oriented documentation, training users, supporting users in their initial product use, and reacting to user feedback. User feedback should be confined primarily to product tuning, configuring, installation, and usability. This phase can range from being very simple to extremely complex, depending on the type of product.

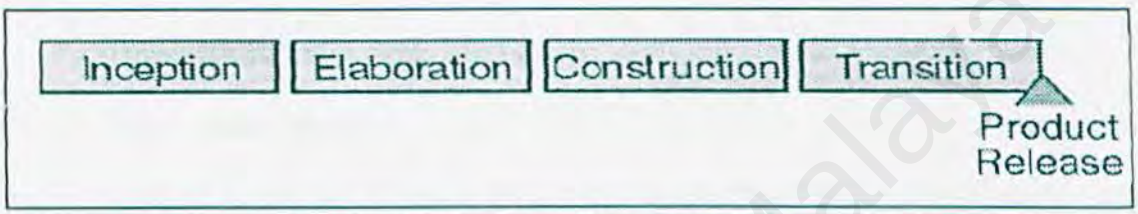


Figure 3.8: Milestone of Transition phase: Product Release

The milestone of transition phase: Product Release Milestone. In some cases, this milestone may coincide with the end of the inception phase for the next cycle.

3.3.2.2 Iterations - The Time Dimension

Each phase in the RUP can be further broken down into a few of iterations. Iteration is a complete development loop resulting in a release (internal or external) of an executable product, a subset of the final product under development, which grows incrementally from iteration to iteration to become the final system. Below are some of the benefits of an iterative approach: -

- Risks are mitigated earlier.
- Change is more manageable.

- Higher level of reuse and better overall of the quality.
- Ensures rapid-delivery and controlled quality at each step in the design.
- Elements are integrated progressively and almost continuously.
- The result is a more robust architecture.
- Testing occurs throughout the entire process.
- Requirements gathering and validation are continuous.
- The design can be separate out from the implementation of a software project.
- The project team can learn along the way.
- The development process itself can be improved and refined along the way.
- Better overall quality
- Enables architects to separate out the design from the implementation of a software project.

3.3.3 The RUP Captures Software Development Best Practices

The Rational Unified Process captures many of modern software development's best practices in a form suitable for a wide range of projects and organizations. Commonly, there are six best practices, which will be depicted in detail in the following.

3.3.3.1 Develop Software Iteratively

Most software teams still use a waterfall process for development projects. This inefficient approach idle key team members for extended periods and defers testing until the end of the project lifecycle. This might pose a serious threat to release deadlines if problems tend to be tough and expensive to resolve. By contrast, RUP represents an iterative approach.

In the RUP, the iterative approach is very controlled; the number, duration, and objectives of iterations are carefully planned, and the tasks and responsibilities of participants are well defined. In addition, objective measures of progress are captured.

3.3.3.2 Manage Requirements

Requirements management is a methodical approach to elicit, organize, communicate, and manage the changing requirements of a software-intensive system or application. The profits of effective requirements management are including: -

- Better control of complex projects
- Improved software quality and customer satisfaction
- Reduced project costs and delays

The RUP is a use-case-driven approach. Use cases used for capturing requirements and defining the behavior of the system. It plays a major role in several of the process workflows, especially design, test, and project management.

3.3.3.3 Use Component-Based Architecture

Use cases drive the RUP throughout the entire lifecycle, but design activities center on architecture. The main focus of early iterations is to produce and validate software architecture. In the initial development cycle, this takes the form of an executable architectural prototype that progressively evolves, through consequent iterations, into the final system.

A component can be defined as a nontrivial piece of software: a module, package, or subsystem that fulfills a clear function, has a clear boundary, and can be integrated into a well-defined architecture. It is the physical realization of an abstraction in the design. The RUP supports component-based development in several ways: -

- The iterative approach allows developers to progressively identify components and decide which ones to develop, which ones to reuse, and which ones to buy.
- Concepts such as packages, subsystems, and layers are used during analysis and design to organize components and specify interfaces.
- The focus on software architecture articulates the structure. The architecture enumerates the components and the ways they integrate, as well as the fundamental mechanisms and patterns by which they interact.
- Testing is organized around single components first and then is gradually expanded to include larger sets of integrated components.

3.3.3.4 Visually Model Software

Models are simplifications of reality; help to understand and shape both a problem and its solution. A large part of the RUP is about developing and maintaining models of the system under development. The Unified Modeling Language (UML) is a graphical language used to gives a standard means of writing the system's blueprints and covering conceptual items such as business processes and system functions. Besides, it provides the vocabulary to express various models.

3.3.3.5 Continuously Verify Quality

In software development, quality is focused on two areas: -

- **Product quality** -- The quality of the principal product being produced (the software or system) and all the elements it comprises.
- **Process quality** -- The degree to which an acceptable was implemented and adhered to during the manufacturing of the product. Additionally, process quality is concerned with the quality of the artifacts produced in support of the principal product.

3.3.3.6 Control Changes to Software

Many work products are often modified in an iterative development. Iterative development emphasizes the vital issues of keeping track of changes and ensuring that everything and everyone is in sync. Change management is a methodical approach to managing changes in requirements, design, and implementation. It also covers the important activities of keeping track of defects, misunderstandings, and project commitments as well as associating these activities with specific artifacts and releases. Change management is tied to configuration management and measurements.

3.3.4 Profits of Rational Unified Process

The RUP has several strengths: -

- It is based on software engineering principles such as an iterative, requirements-driven, and architecture-based approach to development.
- It provides several mechanisms, such as a working prototype at the end of each iteration and the go/no-go decision point at the end of each phase.

- Rational continues to make a significant investment in its RUP product, an HTML-based description of the Unified Process that the organization can tailor to meet its exact needs.

3.3.5 Weaknesses of Rational Unified Process

The RUP suffers from several weaknesses: -

- It is only a development process. As shown in Figure 3.2, it misses the concept of maintenance and support.
- It does not explicitly support multi-project infrastructure development efforts and missing opportunities for large-scale reuse within the organization.
- The process is likely not yet sufficient for the complex needs of modern developers because Rational's approach was initially tools-driven.

3.3.6 Unified Modelling Language

The Unified Modelling language (UML) is a set of standardizes notations that resulted from the brainchild of Grady Booch, James Rambaugh and Ivar Jocobson. Prior to the UML, there was no clear leading modelling language. Many similar modelling languages shared a set of commonly accepted concepts that are expressed slightly differently in various languages.

The UML is a graphical language for specifying, visualizing, constructing, and documenting the artefacts of software-intensive systems. It represents a collection of the best engineering practices that have proven successful in the modelling of large and complete systems.

Develop a model for an industrial-strength software system prior to its construction or renovation is an essential as having a blueprint for large building. Good models are essential for communication among project teams and to assure architectural soundness. As the complexity of systems increase, so does the importance of good modelling techniques.

UML has been design for a wide range of modelling. Meanwhile, the system development focuses on three different models, which are: -

- **Functional model:** describes the functionality of the system from the user's point of view using the use case diagram.
- **Object model:** describes the structure of a system in term of the objects, attributes, association and operations using the class diagram.
- **Dynamic model:** describes the internal behaviour of a system using sequence diagram, state chart diagram and activity diagram in UML.

3.4 Chapter Summary

A software development methodology is very important for a software invention. It is needed to ensure the entire development is always within schedule and budget and deliver high-quality software that meets all the end-user requirements.

CHAPTER 4: SYSTEM ANALYSIS

4.1 Introduction of System Analysis

System analysis is a development of the high level, architectural design for a solution system *[Jacobson et al., 1996]*. It is focus on what the existing system does but not how it will be done. The analysis phase is essential to provide a clear insight into data input, processing and output without considering the end deliverables. A series of activities is carried out for this purpose and eventually leads to a set of system requirement specification. Thus, this process is sometimes known as fact-finding.

4.2 Fact-Finding Techniques

Appropriate fact finding techniques are used to establish and elicit the facts, which are relevant to the design of a new system. The facts discovered form the basis for defining the requirements and functions of the new system. Three techniques are used for this project: documentation, interview and questionnaires.

4.2.1 Documentation

A review of documentations on existing systems is usually a good starting point for fact-finding. Documentations are effective way for manual describing how the current system operates. A good picture of the flow of information usually can be gained through the studies on the various documentations, especially user manuals of the current systems. In additionally, the strengths and the weakness for each current system can be identified. Existing system which had studied are: Traditional Collaborative Learning Approach, Blackboard System, coMentor, and NetOp.

4.2.2 Interviewing

Interviewing is the most predominant fact-finding technique. It has the potential advantage of providing face to face communication with a range of users to verify facts. It also can probe for more feedback and observe the nonverbal communication, which consider is helpful for the information gathering.

A few interview sessions has been conducted with some of the secondary school teachers and students of Faculty of Education in UM, whose are former teachers of primary school. From those sessions, some of the user key requirements and expectation to the new system have been gained.

4.2.3 Questionnaires

Questionnaires are the most efficiency way to collect facts from a large number of people while maintaining standardized responses. It is inflexible method compare to interviews but information can be obtained more quickly and cost effective. Furthermore, responses can be tabulated and analysed quickly. However the risk of the number return can be low.

70 sets of questionnaires are prepared in order to elicit more accurately facts. The target respondents are secondary school teachers, students and lecturers of Faculty of Education in UM. The graphs shown below are the analyzed results based on 40 set of questionnaires that are returned.

Section A of the questionnaires focus on learning effects. The bar chart shows which is the premier criteria for the implementation of CLT, from perspective of teaching-learning process and student's gained. As a result, cooperation is the most important feature that needs to be emphasized in both perspectives.

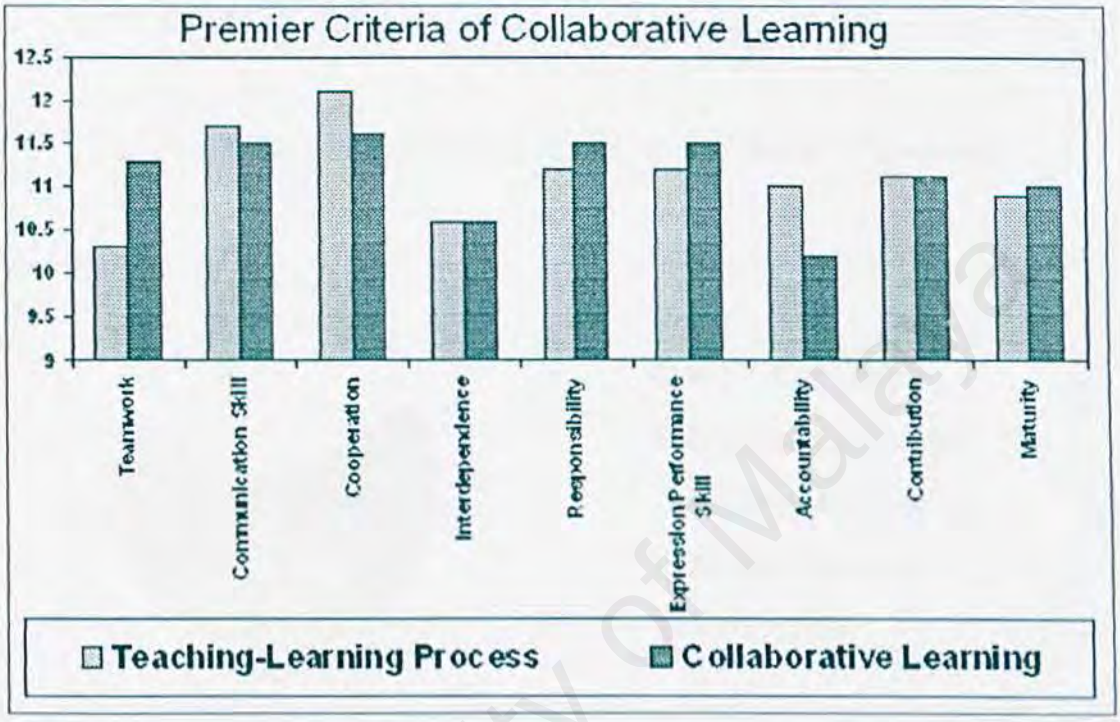


Figure 4.1: Bar Chart of Comparison of Premier Criteria

Section B of the questionnaires concerns the initiative/facilities that are available nowadays to implement collaborative learning. The first pie chart below shows the parties that put a lot of effort in implementing the collaborative learning. Government gains the highest rate while primary and secondary schools are the least. The second pie chart shows the sufficiency of facilities that are provided for collaborative learning purpose. It is fact that high education institutes and enterprise usually provide enough facilities compare to the others.

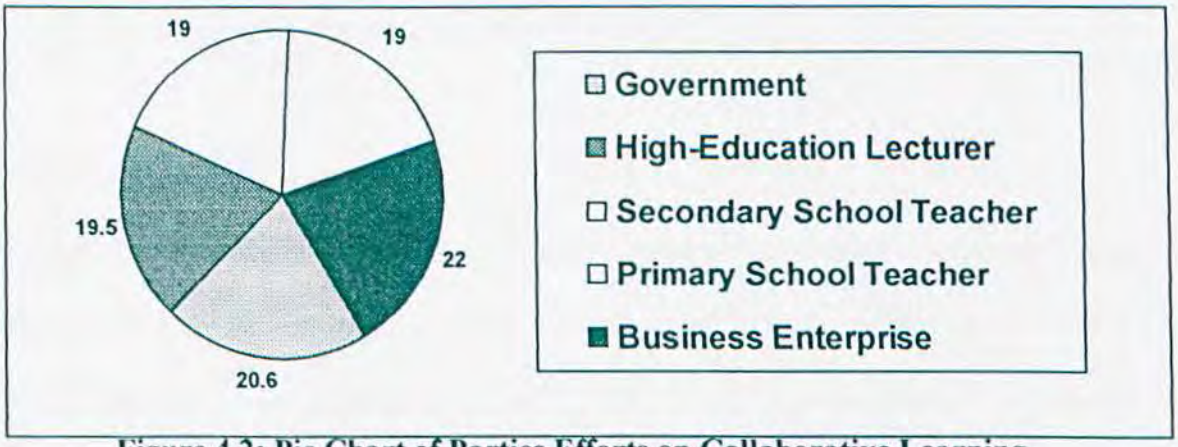


Figure 4.2: Pie Chart of Parties Efforts on Collaborative Learning

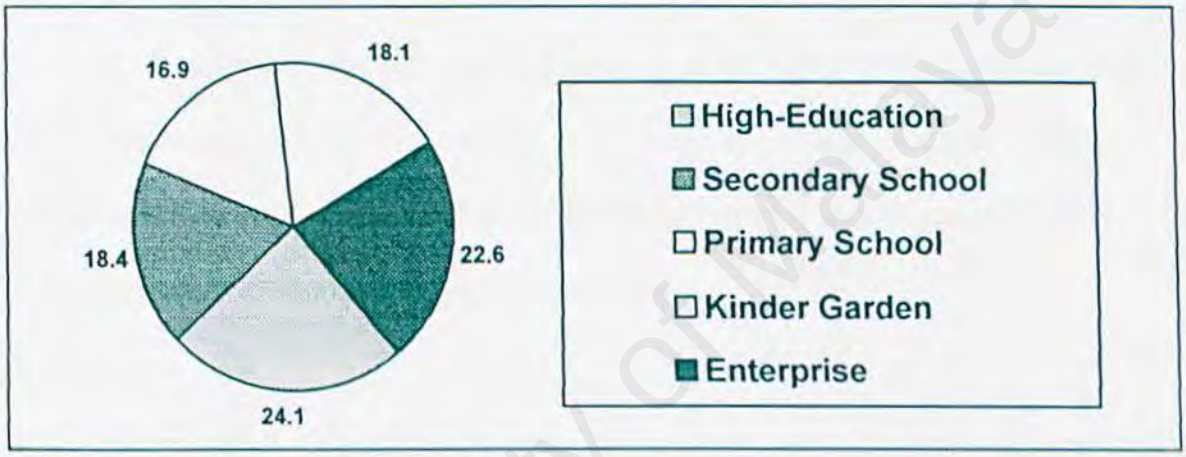


Figure 4.3: Pie Chart of Sufficiency of Facilities Provided by Different Parties

Last part of the questionnaires focus on issues of existing collaborative learning tools. Most of the respondents require the tool should meet the following requirements:

- Provides clear guides and instructions
- Focus on students interactions
- Provide major functions, including automatic grouping, bulletin board and instructor monitoring
- Easy to use
- Avoid too complex in managing

- Avoid too many bureaucracy steps to follow.

4.3 Functional Requirements

Functional requirement is a statement of the services or functions that a system should provide how the system reacts to particular inputs, and how the system should behave in particular situations *[Sommerville, 1998]*. It is a set of functions that are required to be included in the system in order to achieve the objectives of CLT.

CLT contains 3 distinct modules: main module, Jigsaw-Puzzle module and Think-Pair-Share module. This project will only concentrate on the functional requirements for main module. The others can be referred to my group members' (Miss Teo Poh Ling and Miss Teh Hwee See) studies.

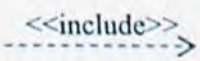
Each of these sections has different roles and use cases. The use case diagram, which is shown in the figure 4.4, is used to describe and determine the overall functional requirements of the CLT.

4.3.1 Use Case Diagram

Use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actor and the uses, and generalization among the use cases *[Ali Bahrami, 1999]*. It is always used to represents the functionality of a system. An actor represents the role of the system's user or an entity that resides outside the system, such as another system or database. While a use case represents a sequence of actions that an actor performs within system

to achieve particular role. The use cases capture all functional requirements of the system. The notation uses in use case diagram are depicted in table 4.1 below.

Table 4.1: Notation of Use Case Diagram

Notation	Description
	*Actor* – the role of the system's user or an entity that resides outside the system.
	*Use case* -- sequence of actions that an actor performs.
	Unidirectional association -- show communication between actor & use cases.
	*Generalization* -- a parent use case defines behavior that its child can inherit.
	*Include relationship* -- one use case explicitly includes the behavior of another use case.
	*Extends relationship* -- a base use case implicitly includes the behavior of another use case.

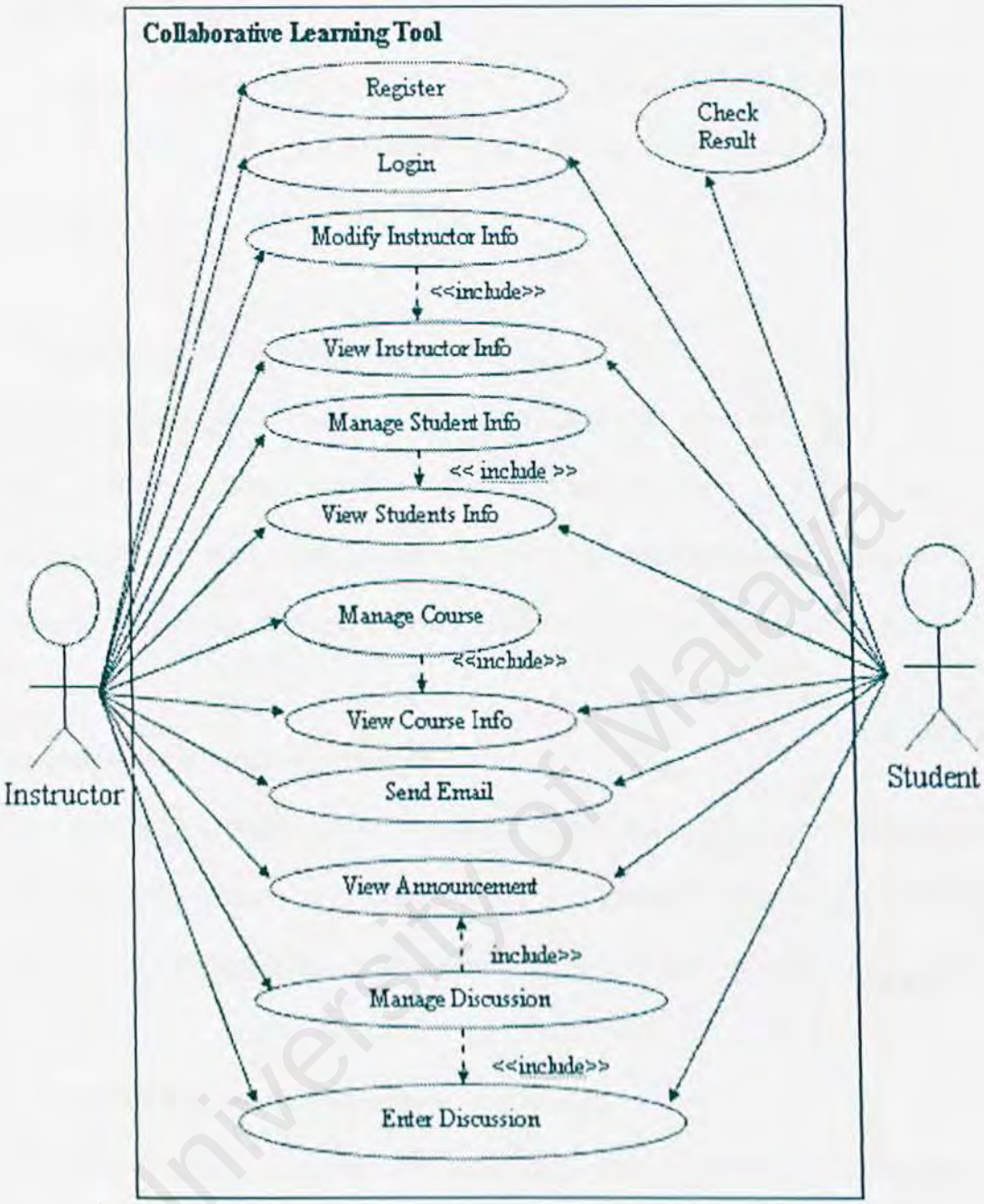


Figure 4.4: Use Case Diagram of Collaborative Learning Tool

CLT consists of two main sections: instructor section and student section. Below are the functional requirements that are defined from the use case diagram in the figure 4.4.

- **Registration Module**

This module will allow the qualified general user to register as an authorized user of the CLT. They are generally called instructor, who has a powerful control right to the tool.

- **Authentication Module**

This module should be able to check the validated user when the user is login to CLT. The tool should not allow the unauthorized users to enter CLT. In additionally, the tool should be able to check for the user's level, instructor or student. Different level of user may have different view and control access right.

- **Instructor Information Module**

This module should only allow the authorized students to view the information of their instructor without other control right. The instructor should be allowed to view, modify his/her own personal information and change his/her login password.

- **Students Information Management Module**

This module should only allow the authorized students to view all the students' information under certain course without other control right. The instructor should be allowed to view all the students' information, add a new student record, modify and remove students' information.

- **Check Result Module**

This module allows the students to check their result for each discussion they participated in. Their grade is given by their instructor depends on their individual and group performance during the discussion and their result from the quiz.

- **Send Email Module**

This module should provide the instructor and student to send email to each other. The instructor has the option either to send to all the students or selected student from a student list.

- **Course Information Management Module**

This module should only allow the authorized students to view the course information and contents, and access the attachments that have been posted by the instructor. Other than that, no further control right is permitted. The system should allow the instructor to create a new course or remove an existing course from the list, view and modify existing course information. In addition, the instructor should be allowed to view, add, modify and remove the course contents/outlines from the time to time. Besides, the instructor should be able to upload relevant information or notes for student references uses.

- **Discussion Management Module**

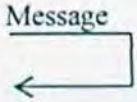
This module should only allow the authorized students to view the announcements about the discussion details (such as title, date and time of the discussion), which are posted by their instructor. They can only enter the discussion session if the date is valid as stated in the announcement. The instructor should be allowed to view,

add, modify and remove discussion from the announcement board. Besides, the instructor also can enter the discussion session any time they like.

4.3.2 Sequence Diagram

Sequence diagram is an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. It shows object participating in the interaction by their life-lines and message they exchange, arranged in a time sequence *[Ali Bahrami, 1999]*. The sequence diagram shows the overall flow of the program's control. The table 4.2 depicts the notations used in the sequence diagram.

Table 4.2: Notation of Sequence Diagram

Notation	Description
	*Object's lifeline* -- represent the object's existence during the interactions.
	*Focus of control* -- shows the period of time during which an object is performing an action, either directly or through an underlying procedure.
	*Message* -- sends a message from an object to another object.
	*Message to self* -- sends a message from one object back to the same object.

The sequence diagrams of the CLT are shown in the appendixes. Each of the sequence diagram is a graphical representation of a scenario of a use case (figure 4.4), and it describes interaction between objects.

4.4 Non-functional Requirements

Non-functional requirements are the constraints under which a system must operate and the standards which must be met by the delivered system *[Sommerwille, 1995]*. These requirements are often considered critical. Failure to meet an individual functional requirement may degrade the system but failure to meet a non-functional requirement may make the whole system unusable. Below are the non-functional requirements for CLT: -

- **Security**

  In web-based application system, security is crucial in order to protect user's privacy and database. CLT deals with the issues of integrity of user registration data, authentically process, and confidential of user information.

- **Inter-operability**

  Computer and application from different suppliers will have the capability to work together on a network and connect to each other. Besides, CLT should provide multi-platform feature which is enabled its website to be launch in most of the web browser engine such as Internet Explorer and Netscape.

- **Robustness**

The system must be able to handle error exceptions that may happen during operation of the system. CLT is designed to have short mean time to failure, low probability of unavailability, low rate of failure occurrence and high availability. Any invalid input must generate a clear and understandable error. The system must check the validity of all entries in a form before accepting the form of submission.

➢ **User-friendly**

Generally, the design of the web-based application interfaces should be friendly and easy to understand. Criteria that are used to build the CLT's web interfaces: -

- *Consistent*. Screen design and error messages displayed are standard in order to reduce confusing and increasing comfort ability.
- *Accommodation* of all kind of level of users.
- *Appropriate error handling* with associated error messages.
- *Easy to use and navigate*. The screens must be easy to read and straightforward.
- *Logical and appealing layout*. The interface must be logical and consistent.

➢ **Efficiency**

Efficiency is understood as the ability of a process procedure to be called or accessed unlimitedly to produce similar performance outcomes at an acceptable or credible speed *[Sommerwille, 1995]*. Efficiency is measured base on response time performance, page generation speed and graphics generation speed.

- **Scalability**

Scalability is a guarantee of the capability of the system to migrate as a client or server to machines of greater or lesser power, depending upon requirements, with little or no change to the underlying components. The solution can be scaled using hardware or application configuration or both.

- **Reliability**

Reliability is the extent to which a program can be expected to perform its intended function with required precision *[Pressman, 2001]*. It is closely related to correct link processing, error recovery and user input validation and recovery. The application system, software and hardware shall be reliable and shall not cause unnecessary and unplanned downtime of the overall environment. Besides, error message should be state clearly for each man-made error.

- **Code Reusability**

Code must be written in the form of reusable, extensible, and well documented. The code must be very readable and well-organized to enhance the reusability in future.

4.5 Development and Runtime Requirements

This part will describe the hardware and software requirements for the CLT, in both development environment and runtime environment.

4.5.1 Hardware Requirements

Table 4.3: Hardware Requirements

	Development Environment	Runtime Environment
Hardware Requirement	• 200 MHz Pentium Processor • 256MB RAM • 40.0 GB Hard Disk • Standard input/output devices (CD-ROM, Modem, mouse, keyboard, printer and etc)	• Pentium III and above computer • 64MB RAM and above • 6.4GB Hard Disk and above • Standard input/output devices (CD-ROM, Modem, mouse, keyboard, printer and etc)

4.5.2 Software Requirements

Table 4.4: Software Requirements

	Development Environment	Runtime Environment
Software Requirement	• MS SQL Server 2000 • Windows 2000/XP • Internet Explorer 5.0 • Netscape Navigator 4.7 • Ms FrontPage 2000 • Personal Web Server • Internet Information Server • Adobe Photoshop 6.0 • Microsoft Visual Studio.Net	• Internet Explorer 5.0 and above • Netscape Navigator 4.7 and above • Ms SOL Server 2000 • Internet Information Server • Personal Web Server (if the operating system is Windows 98 and below) • Any Windows Operating System

4.6 Chapter Summary

Throughout this chapter, very detailed descriptions on needed requirements are performed. These include elaboration on system analysis, fact-finding methods such as documentation, interviewing and questionnaires, and some work in analyzing the result of those fact-finding techniques. After all, by using object-oriented analysis, use case diagram is drawn and several system functional requirements are defined. Some nonfunctional requirements had also been recognized to be important to CLT successful.

CHAPTER 5: SYSTEM DESIGN

5.1 Introduction of System Design

The design phase is the stage, which translates the requirements into the module characteristics. It concerns on how the system functionality is to be provided by the different components of the system. The system design comprises of system functionality design, database design and the user interface design. The first task in system design is to specify application architecture.

5.2 System Architecture

Collaborative Learning Tool (CLT) is accessible using standard internet protocol. CLT are accessed via web protocol and data formats such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). In a raw draft, the CLT architecture consists of a group of end users, in this instance the instructor, the students and the general users. All data needed by each module are stored in same database. It is ideal to thought of CLT as a set of services that are provided to clients that makes use of these services and each of these services will require data stored in one location. The CLT architecture is found suited to the client-server architecture.

5.2.1 Client-Server Architecture

In a client-server architecture, an application is modeled as a set of services that are provided by severs and a set of clients that use these services. The design of client-server system should reflect the logical structure of the application that is being developed. The concept of client-server architectures have already been introduces in Chapter 2.

5.2.2 Three-tier Architecture

Among several types of client-server architectures, the three-tiered architecture is found to be the best suited to CLT. The use of three-tiered client-server architecture allows the information transfer between the web server and the database server to be optimized. The communication between these systems does not have to be based on the Internet standards but can use faster, lower-level communications protocols. Efficient middleware that supports database queries in Structured Query Language (SQL) is used to handle information retrieval from the database.

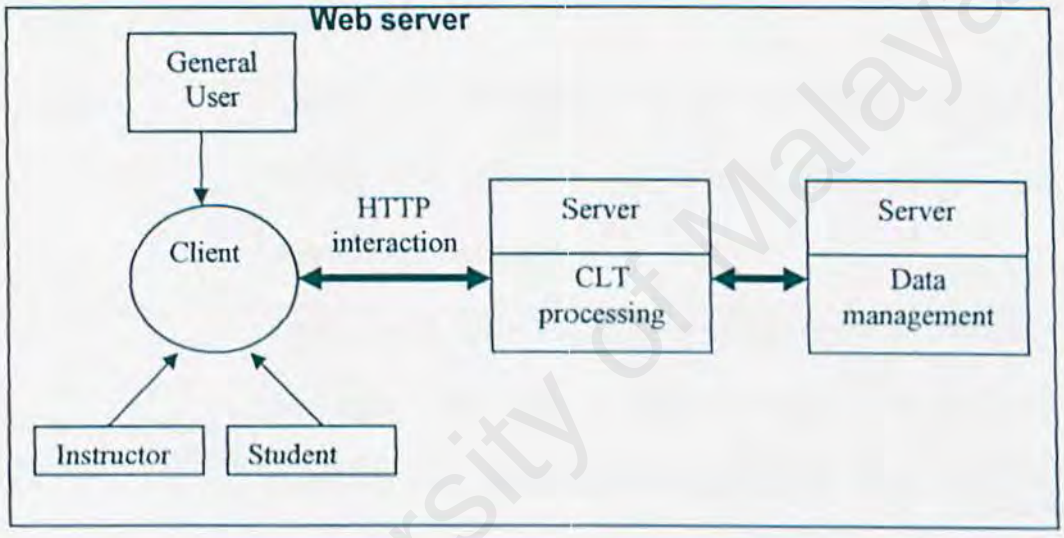


Figure 5.1: Three-tier client-server architecture of CLT

5.3 System Functionality Design

In this phase, CLT is designed from a functional viewpoint, starting with high level view and progressively refining this into a more detailed design. Collaborative diagram and activity diagram will be used in this stage.

5.3.1 Collaboration Diagram

Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. A collaboration diagram shows a set of objects, links among those objects, and messages sent and received by those objects. It also can be used to illustrate the dynamic view of a system. The notation used in the collaboration diagram is depicts in the table 5.1 while the collaboration diagrams of the CLT are enclosed in the appendix B.

Table 5.1: Notation of the Collaboration Diagram

Notation	Description
:Object	Object – typically named or anonymous instances of classes. It may also represent instances of other things, e.g., components and nodes.
———	Links – to show the links/relationship between the objects.
⟶	Messages – to describe the interaction between the objects.
Notes	Note – to give an additional description; it can be put on the certain place which thinks there a need.

5.3.2 Activity Diagram

An activity diagram is essentially a flowchart, showing flow of control from activity to activity. It is often used to model the dynamic aspects of a system. Besides, activity diagram can also model the flow of an object as it moves from state to state at different points in the flow of control. The notation used in the activity diagram is depicts in the table 5.2.

Table 5.2: Notation of the Activity Diagram

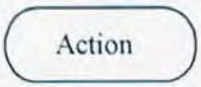
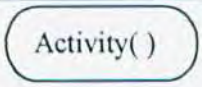
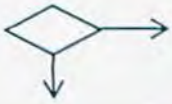
Notation	Description
	Initial state -- the start state of the activity diagram.
	Final state -- the end of state of the activity diagram.
	Action States -- each state representing the execution of an action, cannot be decomposed and atomic
	Activity States -- states can be decomposed; their activity being represented by other activity diagrams; not atomic.
	Branch -- specifies alternate paths taken base on Boolean expression.
	Forking and joining -- one activity can decompose into two activities that can done concurrently and joining back together after all the activities have been done.
	Swimlanes -- separate/partition the activity state on an activity diagram into groups, each group representing the business organization responsible for those activities.
	Shows the flow of the next activity/action state.

Figure 5.2 and figure 5.3 below show the activity diagram of the CLT for each level of users (Instructor and Student).

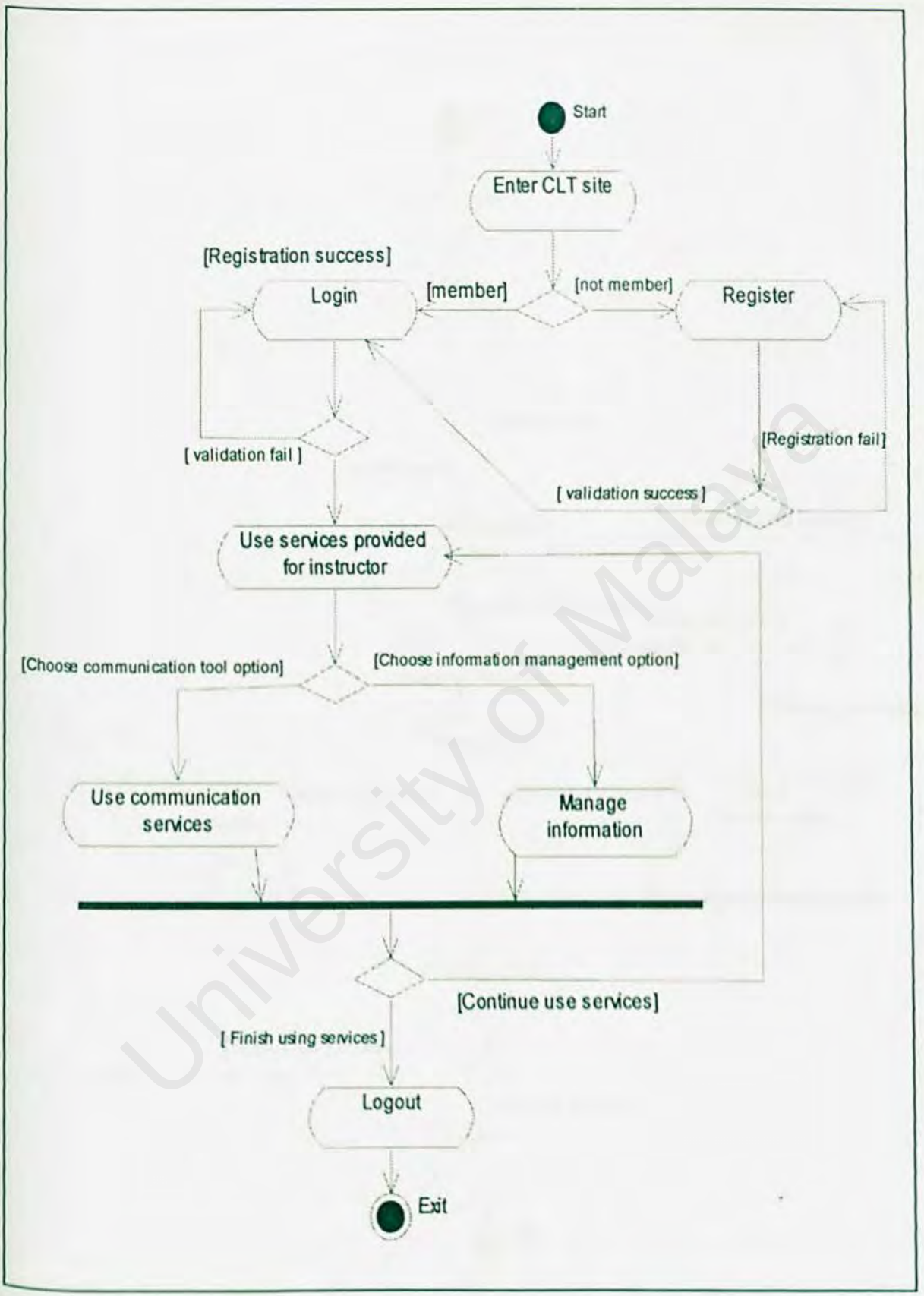


Figure 5.2: Activity Diagram of Collaborative Learning Tool (for Instructor)

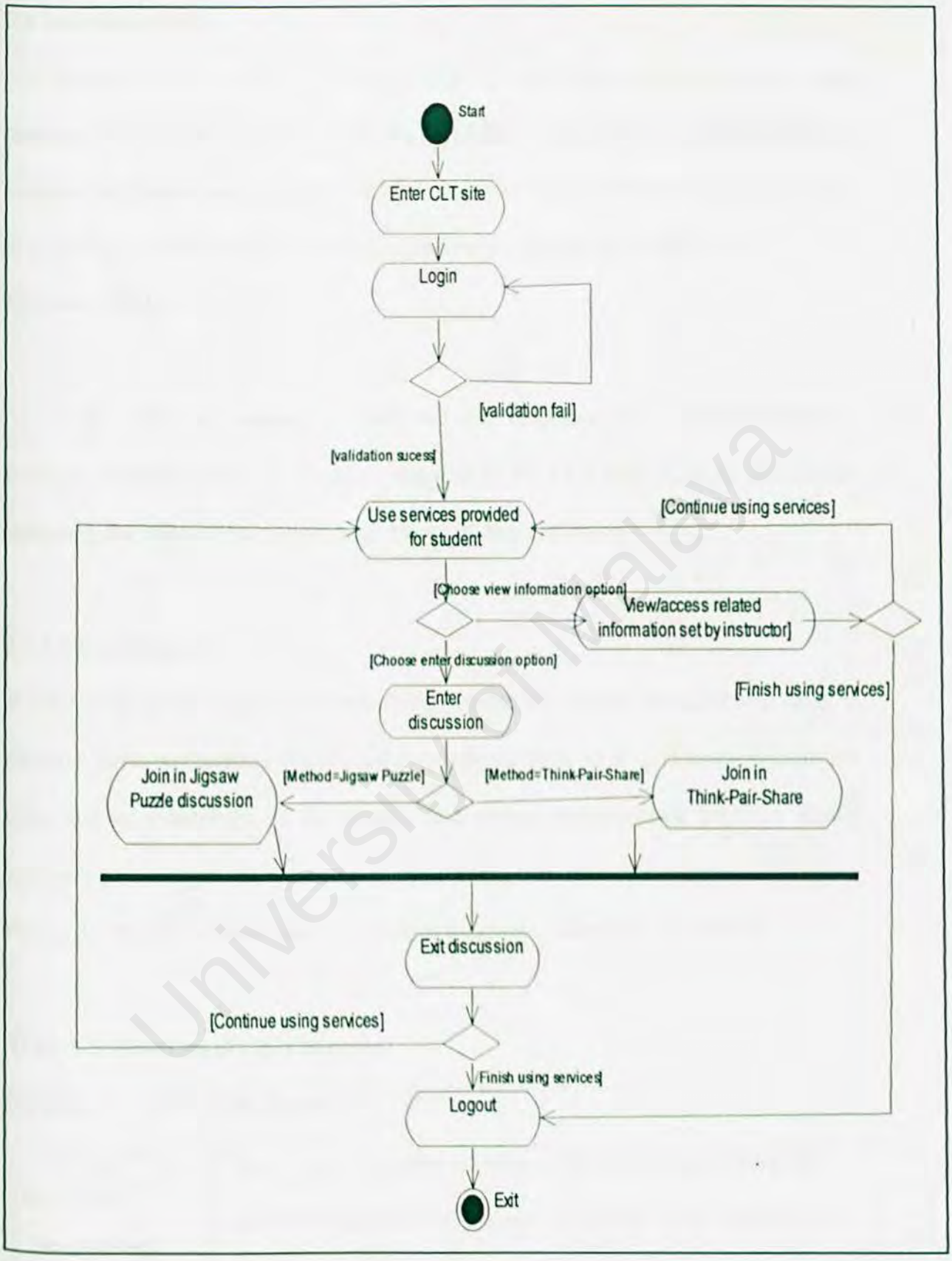


Figure 5.3: Activity Diagram of Collaborative Learning Tool (for Student)

5.4 Database design

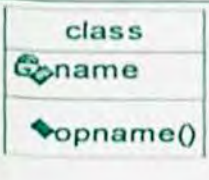
The database design in CLT is based on the relational data model. In a relational database, the schema is made up of tables, consisting of rows and columns, where each column has a name and a simple data type. In object model, the counterpart to a table is a class (or classes), which has a set of attributes (properties of data members). *[Ali Bahrami, 1999].*

The following section is explained the mapping from object-oriented to relational database based on the class diagram of the CLT main module. All details regarding the tables in the database are stored in data dictionary.

5.4.1 Class Diagram

A class diagram is a picture for describing generic description of possible system. It contains icons representing classes and their relationships. It is used to show common roles and responsibilities of the entities that provide the system's behavior during analysis; and capture the structure of the classes that form the system architecture during design. The notation used in the class diagram is depicts in the table 5.3.

Table 5.3: Notation of Class Diagram

Notation	Description
	*Class* -- the class name is shows in the first compartment, the second compartment list the attributes of the class, and the last compartment lists the operation of the class.

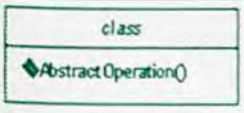
	*Abstract class* -- is a base class, defining operation and state that will be inherited by subclass. An abstract class has no instances.
	*Generalization* -- a parent class defines behavior that its child can inherit.
	Association -- relationship between 2 classes with the multiplicity shown on both end of the line. Multiplicity can be 1 or many (*), many to many and so on.

Figure 5.4 shows the class diagram of the CLT. It shows the interaction between structural elements found in the system. It is based on the scenario-based analysis results (shown in use case diagram and sequence diagrams), which helps to identify objects, attributes and operations in the system.

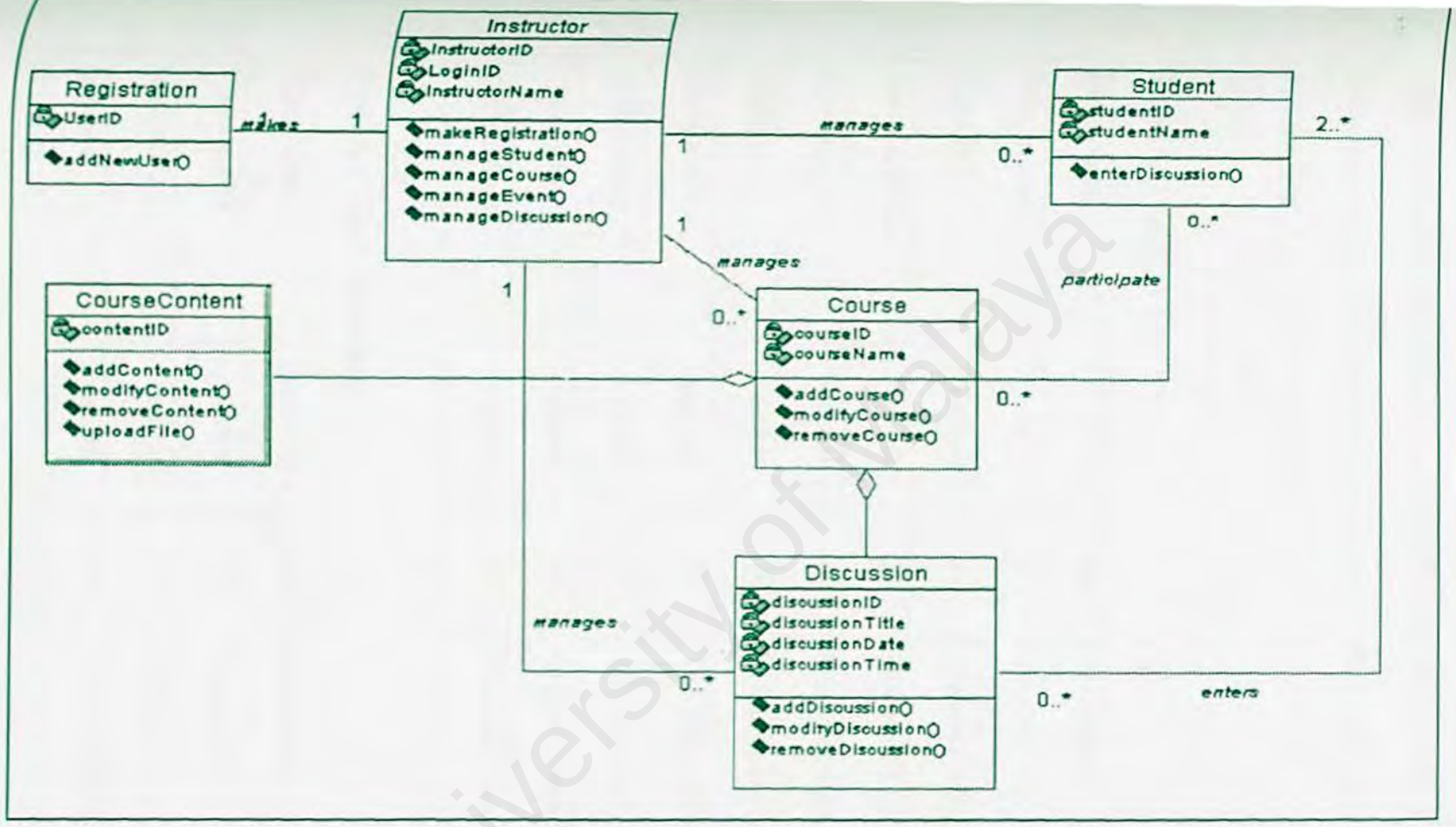


Figure 5.4: Class Diagram of Collaborative Learning Tool

5.4.2 Data Dictionary

Data dictionary describes the field name, field type, field length and the description of each field in all tables stored in the CLT database. There are 10 tables of data dictionary stored in the database. For each table, the primary key is shown with (**) and the foreign key is shown with (*).

Table 5.4: Table of Instructor

Field Name	Field Type	Length	Description
**InstructorID	int	4	Instructor ID
LoginID	Varchar	20	Instructor's login ID
InstructorName	varchar	50	Instructor's name
Password	varchar	10	Instructor login password
Address	varchar	100	Instructor address
InstituteName	varchar	50	Teaching institute name
InstituteAdd	varchar	100	Teaching institute address
EmailAdd	varchar	50	Instructor email address
ContactNum	Number	15	Instructor contact number

Table 5.5: Table of Student

Field Name	Field Type	Length	Description
**StudentID	varchar	8	Student ID
StudentName	varchar	50	Student name
Address	varchar	50	Student address

StudentContactNum	Number	15	Student contact number
StudentEmailAdd	varchar	50	Student email address
Gender	varchar	2	M=Male; F=Female
Race	varchar	2	M=Malay; C=Chinese; I=India; O=other
Ability	varchar	2	Student ability in the course
InstructorID	int	4	Instrutor ID

Table 5.6: Table of Course

Field Name	Field Type	Length	Description
**CourseID	varchar	8	Course ID
CourseName	varchar	30	Course name
CourseDescription	varchar	50	Course description
DateCreated	Datetime	8	Date course is created
StudentLoginPwd	varchar	20	Password assigned by instructor for student login uses
*InstructionID	int	4	Instructor ID

Table5.7: Table of CourseContent

Field Name	Field Type	Length	Description
ContentID	int	4	Course content ID
ContentTitle	varchar	50	Content title
ContentDescription	varchar	250	Content Description

AttachFile	varchar	50	Attach file name
CourseID	varchar	8	Course ID

Table 5.8: Table of StudentCourse

Field Name	Field Type	Length	Description
*StudentID	varchar	8	Student ID
*CourseID	varchar	8	Course ID

Table 5.9: Table of Discussion

Field Name	Field Type	Length	Description
**DiscussID	int	4	Discussion ID
DiscussTitle	varchar	50	Discussion title
DiscussDescription	varchar	300	Discussion description
Method	varchar	3	JP=Jigsaw Puzzle; TPS=Think-Pair-Share
DiscussDate	Datetime	8	Discussion date
DiscussTime	Datetime	8	Discussion time
CourseID	varchar	8	Course ID
Phase	varchar	6	Discussion phase
Status	varchar	5	Discussion current status
UploadDocName	varchar	50	Upload document name
DocDate	datetime	8	Upload document date

Table 5.10: Table of StudentDiscussion

Field Name	Field Type	Length	Description
**StudentID	varchar	8	Student ID
**DiscussID	int	4	Discussion ID
Mark1	int	4	Evaluate mark
Mark2	int	4	Evaluate mark
Mark3	int	4	Evaluate mark
Mark4	int	4	Evaluate mark
Grade	varchar	2	Student grade in each discussion
SendTextAbility	varchar	3	Send text controller
DrawingAbility	varchar	3	Drawing ability controller
ExpertGroupID	varchar	8	Expert group ID
NormalGroupID	varchar	8	Normal group ID
PairID	varchar	8	Pair ID
LoginStatus	varchar	5	Student login status
UploadAnsName	varchar	50	Upload answer name

5.5 Graphical User Interface Design

The interface of a system works as a central communication between the processing functions and the user requests. The objective of an interface is to enable the user to grab information that they need or to act as a medium for them to supply more information to the system. The interface is aimed to improve efficiency and

effectiveness of the user when using the entire system. In addition, a good interface shall not cause the user to remember a large number of commands or codes, but should as user-friendly as possible. Below is the prototype for a user interface.

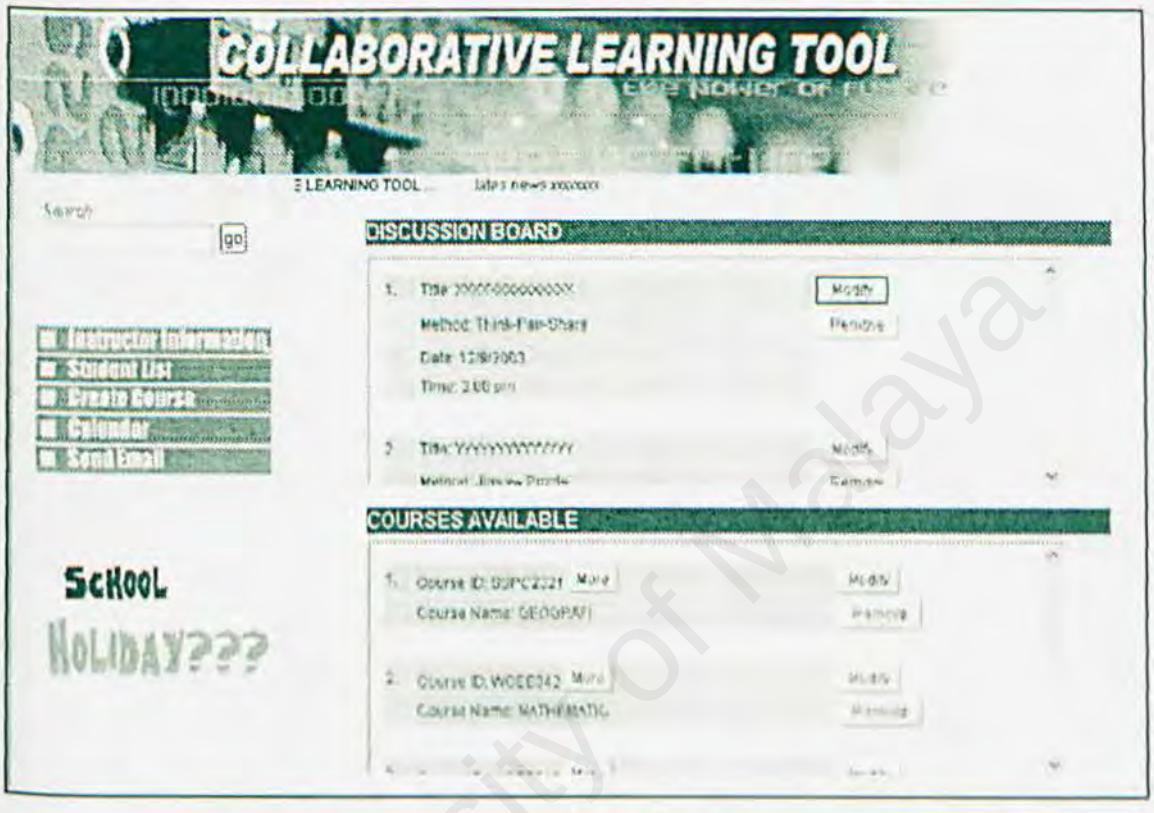


Figure 5.5: User Interface of Collaborative Learning Tool

5.6 Chapter Summary

The system designs require experience as well as systematic planning. This is due to only with the proper system design; the project can be carried out faster and easier. In addition, it will be used to build source codes in order to implement the design generated from before.

CHAPTER 6: SYSTEM IMPLEMENTATION AND DEVELOPMENT

6.1 Introduction of System Implementation

System implementation in a nutshell is the construction of the application. The initial stage of system implementation involves setting up the development environment. This includes setting up development tools to facilitate the system implementation. Process to convert system requirements into program codes is done in this phase. Generally, this phase always involves some modifications to the previous design due to the limitation of the programming language used.

6.2 Development Environment

The usage of dynamic and suitable hardware and software can help accelerate the development or construction of any system. The following sections discuss the hardware and software tools used to develop and document CLT.

6.2.1 Hardware Development Environment

The hardware configuration of the CLT development environment is listed below: -

- 200 MHz Pentium Processor
- 256 MB RAM
- 52x CD-ROM Drive
- 40.0 GB Hard Disk Drive
- Others standard desktop PC compliance

6.2.2 Software Development Environment

The following table summarized the software tools used in the development of CLT: -

Table 6.1: Software Tools Used for the Development of CLT

Software	Usage	Description
Microsoft Windows XP Professional	Development Environment System Requirement	Operating System (OS)
Microsoft Visual Studio .NET Framework	System Development	Development tools for coding the web pages
Microsoft Internet Information Service	Web Server	Web Server Host
Microsoft SQL server 2000	Database Design	Database design, construction and implementation for data storage and manipulation
ASP.NET	System Development	Programming language to coding web pages
VB.NET	System Development	Programming language to coding web pages
Hyper Text Markup Language (HTML)	System Development	Coding the web pages
Adobe Photoshop 7.0	Interface Design	Creating CLT logo and button
Swish 2.0	Interface Design	Creating animation in main page
Macromedia Dreamweaver MX	User Interface Design	Designing the header of the web pages
Internet Explore 6.0	System Development	Web Browser for running the system
Microsoft Word 2000	Documentation	Document the report and user manual.

6.3 System Development

The following sections describe on how to implement and develop the CLT, including configuration of database server, system coding and user interface design.

6.3.1 Database Implementation

Microsoft SQL server 2000 was chosen as a database server in the development of CLT. It is used to keep users' details. CLT is an online application, which the instructor can create, edit and delete any records directly into the CLT database (Figure 6.1). The implementation of the database is described briefly as below: -

- Creating a new database for CLT using Enterprise Manager, named CLT1
- Design tables as stated in system design and name the tables
- Insert data name, data type and length of the data for each table (Figure 6.2)
- Manage relationship of the tables in a diagram form (Figure 6.3)
- Insert default data in the database

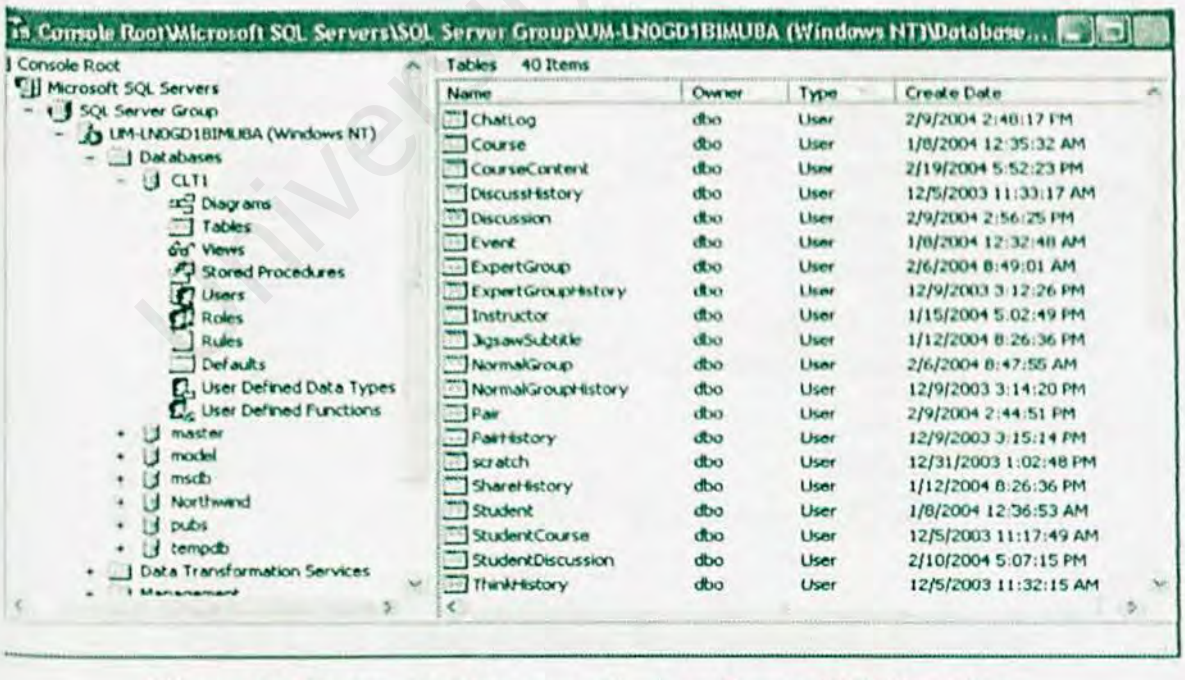


Figure 6.1 Enterprise Manager Used to Create CLT Database

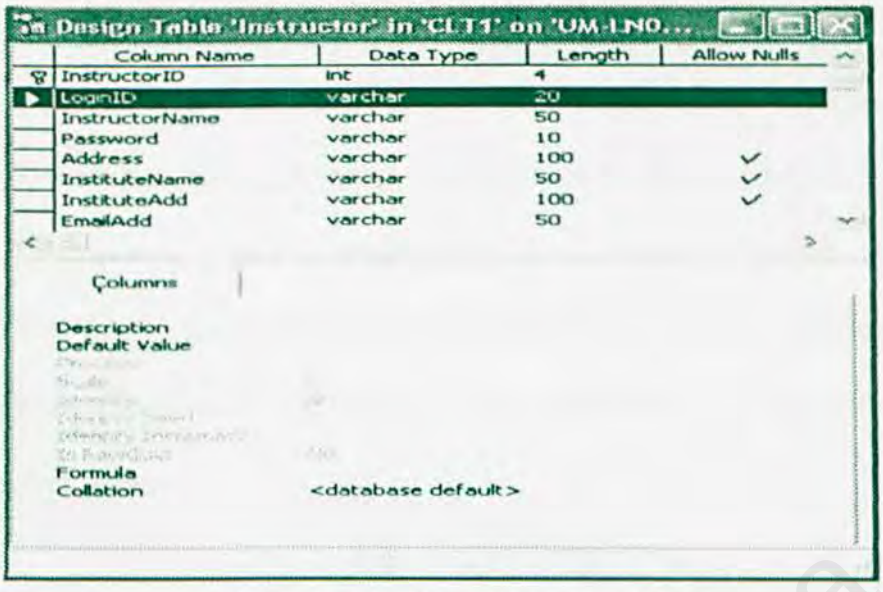


Figure 6.2 Design Table of CLT

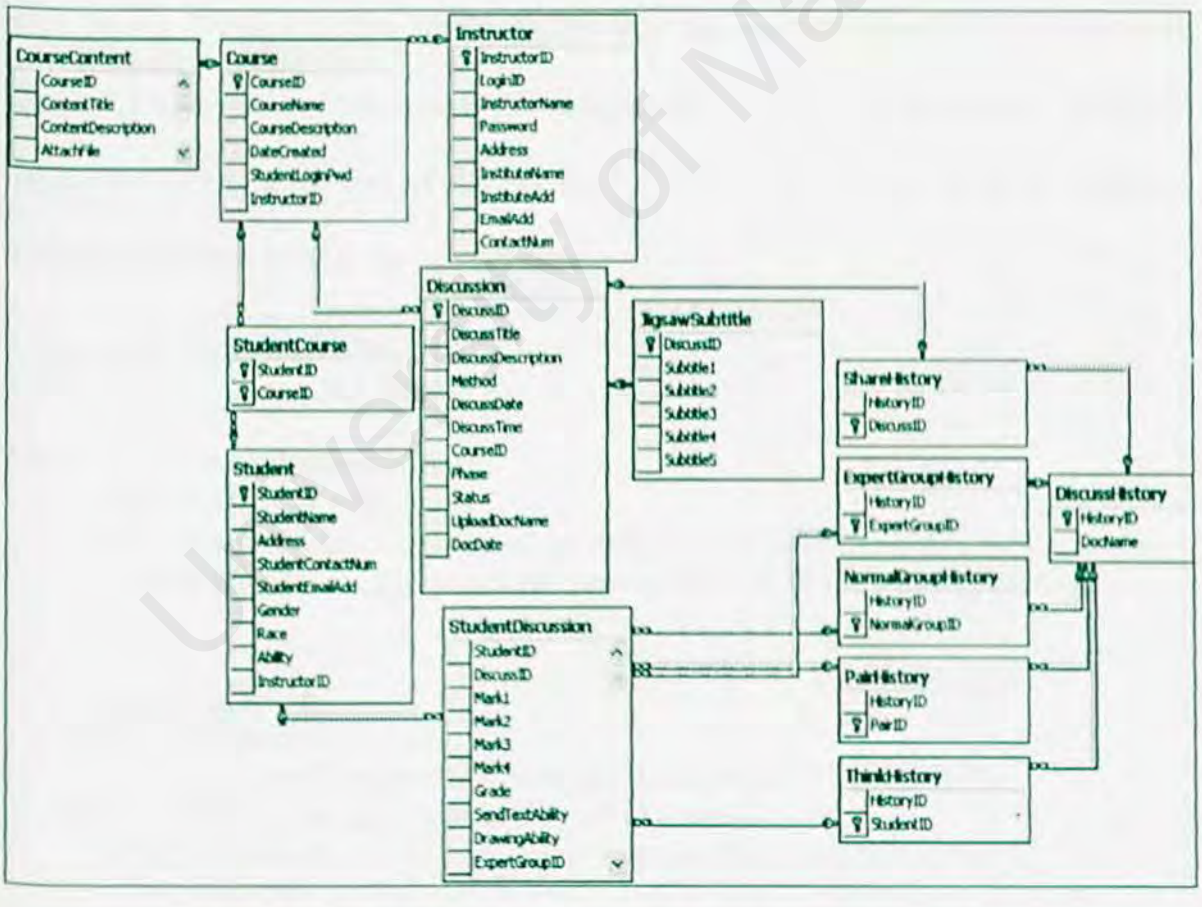


Figure 6.3 CLT Diagram Show Relationships of the Tables

6.3.2 System Coding

Coding is a process that translates a detail design representation of software into a programming language realization which is in machine-readable form. Microsoft Visual Studio .NET Framework is used in creating all of the modules. In the process of coding, every algorithm that is designed during design phase is transformed into lines of code. Comments were written within the code to aid the understanding of the coding and ease for future enhancement.

6.3.2.1 Connection to Database

VB.NET Framework provides several types of database connectivity engines for Access, SQL Server, Oracle and other databases. For SQL Server, each module in the framework that needs to access and manipulate the database has to import "System.Data.SqlClient". Figure 6.4 below show part of the coding that set up the connection to the database server to read data by SQL query statement.

```
Imports System.Data
Imports System.Data.SqlClient

Public Class Announcement
    Inherits System.Web.UI.Page
    Protected WithEvents lblInform As System.Web.UI.WebControls.Label
    Protected WithEvents dgAnnoun As System.Web.UI.WebControls.DataGrid
        :
        :
    'Set up connection
    Dim myConnection As New
       SqlConnection(ConfigurationSettings.AppSettings("DBconn"))
    Dim mySQL As String = "Select * From Discussion"
    Dim myDataReader As New SqlDataReader(mySQL, myConnection)
        :
        :
  End Class
```

Figure 6.4 Part of Source Code for Connecting to the Database Server

Besides, web configuration in the .Net framework of CLT project also needs to be setup its connectivity to the proper database server, as shown in the figure 6.4. This can ease for any change of database server in future as only need to modify the code once in this configuration.

```
<configuration>

    <appSettings>
        <add key="DBconn" value="server=(local); database=CLT1; uid=sa;
        password=;" />
    </appSettings>

</configuration>
```

Figure 6.5 Setting the web.config to Connect to SQL Server

6.3.2.2 Coding Approach

The coding approach used in the development of CLT is the top-down approach. Besides, bottom-up approach is also applied for few of the system coding (source code of the CLT will be shown in appendixes). Below is the briefly explanation on these two coding approaches being used: -

- ***Top-down Approach***

  In top-down approach, modules to be accomplished is broken down into sub modules and then further decomposed into smaller sub modules or functions, and so forth. All these small modules or functions are built and developed separately. By using this approach, CLT is decomposed into several modules (e.g. Instructor, Student, Course etc). Each of these sub modules is then decomposed into sub modules or function (e.g. Add, Edit or Delete record).

- ***Bottom-up Approach***

  Lower level modules will be constructed first before the higher-level modules. The higher modules are just the skeletons that call the lower modules. Once the modules and sub modules under the lower modules are coded, higher modules will be creates to link each of the modules together.

6.3.3 System Debugging

System bugs are a natural part of the development process; as human-being is simply error-prone. Therefore, debugging should be embraced as a necessary part of the development process in order to track and correct program bugs. Two fundamental debugging strategies used are bug prevention and bug detection.

6.3.3.1 Bug Prevention

Bug prevention is the process of eliminating the occurrence of bugs before they have a chance to surface. It is treated as a primary way to eliminate bugs by applying some prevention codes in the programs. Exception handling is one of the useful prevention debugging mechanism. It can detect and respond to unexpected events during runtime by its try and catch clause.

A try clause tells the runtime system that a section of code could cause trouble. Another piece of code (a handler) is needed in a corresponding catch clause, which responds to errors cause by the code in the try clause. Figure 6.6 below show an exception handling in CLT code which inform user that the upload file is failed for certain reason.

```
Try
    'Save the uploaded file to the server.
    tfFileAttach.PostedFile.SaveAs(strFilePath)
    lblUploadStatus.Text = strFileName & "has been successfully uploaded."
    strUploadStatus = "PASS"

Catch Ex As Exception
    lblUploadStatus.Text = "Upload File failed for the reason: " & Ex.Message
    strUploadStatus = "FAIL"

End Try
```

Figure 6.6 Exception handling in CLT Code

6.3.3.2 Bug Detection

The system will still be contended with a certain number of bugs even bug avoidance technique has been applied. Hence, program stepping, breakpoint and watch window is used to track down the bugs. There are methods that provided by the .Net Framework to step over the coding line by line for each time an event is triggered. As a result, the cause of the error can be easily detected and corrected.

6.3.4 User Interface

The user interface is the only portion of the application that is responsive user interaction. It should include all event-handlers or events in response to user request (e.g. click, mouse over, page change etc). It also includes those procedures that either fill controls with data or retrieve data from controls. In short, user interface layer should perform the following tasks: -

- Displaying all application data or information via windows object.
- Initiating all user request (e.g. input, modify and delete data via window object).
- Responding to the changing states of window object.

6.4 Program Optimization

The development of CLT involves many database manipulation, user-defined functions and integrated sub-modules. Hence program optimization is needed to improve its efficiency and effectiveness. There are two ways to implement program optimization: -

- ***Increase the execution speed of the program***
  - Avoid using variant data types, which requires additional internal program standards to identify the information being stored.
  - Minimize the amount of program initialization (inside Page_Load event) which forced to appear before startup code is executed.
  - Breaking codes into functions and sub-functions. Repeating actions can be stored in functions to save memory and increase execution effectiveness.

- ***Decrease the program size (amount of memory use)***
  - Reviewing codes for unused variants, constants and "dead-code" and remove them from the program codes.
  - Assigning the string variables to zero-length string if it is no longer needed.

6.5 Chapter Summary

System implementation is an important phase to ensure the successful transmission of system design and system requirements to real product for real world usage. During the implementation phase, system debugging is being carried out occasionally to prevent any of the system bugs. Testing should be done to test the accuracy and reliability of the application. Testing of the system will be discussed in the following chapter.

CHAPTER 7: SYSTEM TESTING

7.1 Introduction of System Testing

System testing is a critical phase of ensuring its quality control and assurance. Testing represents the complete and extensive review and challenge on the application design, specifications and codes. Its main objective is to identify the errors and run-time program bugs as much as possible and to eliminate them.

The testing process should proceed in stages where testing is carried out incrementally in conjunction with system implementation. In general, the testing process of CLT can be shown in the Figure 7.1. All the details will be further explained in subsequent sub-sections.

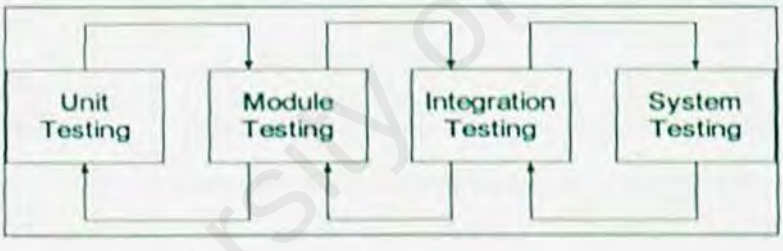


Figure 7.1 Testing Process

7.2 Unit Testing

Unit testing aims at the verification of the smallest unit within a program. Each unit was tested independently to assure accuracy. In CLT, each module contain sub module, which in turn consist of difference functions and units. These functions were individually tested before the entire application was tested. The white-box and black-box approach was used to carry out unit testing.

7.2.1 White-box Testing

White-box testing, sometimes also called glass-box testing uses the control structure and implementation of the procedural design to derive test cases. Tests are conducted, whereby lines of codes are examined one by one to ensure that small unit of codes are in accordance with system specifications. The white-box testing for CLT has been divided into several coverage categories as shown in Table 7.1 below.

Table7.1 Coverage Categories of White-box Testing

Coverage Categories	Explanation
Segment coverage	Every segment of the code between control structures is supposed to execute at least once.
Branch node coverage	Every branch at every possible direction is taken at least once.
Compound condition coverage	When multiple conditions appear in the code, every possible combination is tested based on a truth table.
Data flow testing	Data flow testing is meant for reflecting dependencies that are mainly caused by sequences of data manipulation.
Loop testing	Loop testing is difficult to test when dependencies exist among the loops or between a loop and the code it contain.
Basic path and path testing	Each independent path throughout the code is usually taken at a predetermined order. When dependencies appears in the code, each path where dependency appears exists must be tested.

7.2.2 Black-box Testing

Black-box testing, also called behavioral testing, relies on the specification of the system or software. This testing includes error guessing and boundary value analysis to find incorrect or mission functions, interface errors, errors in data structures, performances errors and initialization and termination errors. Some mistakes or values (out of acceptable range) are entered to test the system reaction and determine whether the system will prompt out the appropriate error message. Besides, it also tests the functionality of the system in an ad hoc basic without knowing the logic structure of the code. Input is provided and output is verified manually to check for accuracy.

7.3 Module Testing

A module consists of a collection of dependent components to perform a particular task or function. In module testing, units in different modules are tested together to ensure the flow between the codes of the modules are not disrupted. Different possible test cases are applied to the module and the test results would be verified. The modules are tested with some dummy data. If an error occurs, the related error to that unit is checked and then modified. Testing is repeated until no error occurs.

7.4 Integration Testing

Integration testing begins after all objects, components and individual modules have passed local unit tests. System with integrated modules must go through integration testing to ensure valid linking and dynamic relationship establishments between modules and sub modules of the whole system.

Every link to all modules was tested and all components such as variables passing, parameters passing of function and event procedure calls, inter-module variables and control values passing must be tested again in the integration testing. The flows of the information between modules are also validated for accuracy and completeness in this testing phase.

7.5 System Testing

System testing is designed to reveal bugs not possible attributed to individual components or to interact between components and modules. System test is carried out on the entire integrated system, whereby integrating main module (this project), Think-Pair-Share module (developed by Teh Hwee See) and Jigsaw Module (developed by Teo Poh Ling) as one unit.

System testing activities include a series of different test designed to fully exercise the system to uncover its limitation and to measure its capabilities. It is purposely test for system performance, reliability, accuracy and other criteria. Besides, it also concerned with validating that the system fulfills its functional and non-functional requirements.

7.6 Interface Testing

The system interface should be user-friendly and not confusing. It is essential to ensure the user understanding what he or she is doing and what is the expected outcome. Instructions must be given in an appropriate manner and at the appropriate time. Error

messages should be clear and straight to the point. However, the error messages should not bring on any bad feelings on the user or leaving user discouraged to use the system.

Besides, the interface design should not lead the user to key in invalid entries. Data type like data format and data length will be controlled to avoid unnecessary problem. For every input text, the maximum length must be set according to the setting in the database. For example, the password must create in maximum length 8 characters. If over the maximum length, user will be requested to re-enter a new password with required length.

7.7 Chapter Summary

The overall system testing yields the expected results. In the next chapter the conclusion of the whole system is being carried out. The obstacles faced during the development of the system, the system strengths and limitation, and future enhancements are discussed in detail.

CHAPTER 8: EVALUATION AND CONCLUSION

8.1 Introduction

After complete the system testing phase, evaluation of the CLT was carried out. Evaluation is the ultimate phase of developing a system before delivery the system to the targeted end users. It was related to target user environment, attitudes, information priorities and several other concerns that are to be considered carefully before effectiveness can be concluded. The purposely to evaluate the end system is to find out the system strengths, limitations, and it future enhancements. Besides, the problems that were encountered during the progress of the system and its solutions also wil! be discussed in this chapter.

8.2 System Evaluation

After the CLT had been completed for the first, a survey form in questionnaires is distributed to target users for evaluating the tool performance and usability. Then after the second modified of the CLT, especially focus on the interface designing, a survey is conducted with the same questionnaires again. Result is being analyzed. Below is the bar graph showing the comparison among two version of the CLT.

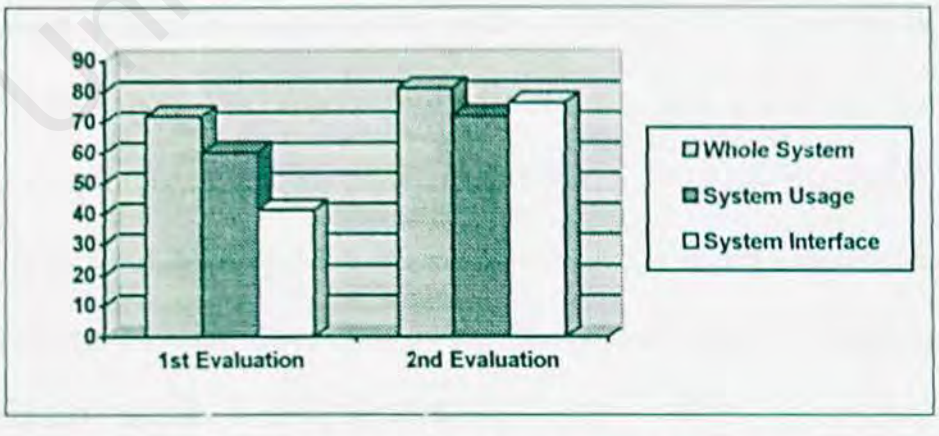


Figure 8.1: Bar Graph of Comparison between First and Second Evaluation

From the bar graph, we can see that the tool performance has been increase dramatically after the modification, especially the system user interface. The newest version with adding some suitable animation in the main page has attracted the general user eye-focus. User input form layout also had been modify to suite into a window without too much of scrolling that are needed. Besides, some of the functions also been added compare to the first version after gathering the new opinion from the target users from the survey conducted. This has successfully increased the evaluation of the new version of CLT.

8.3 Problems Encountered and Solution

Various problems were encountered throughout the development of CLT. Most of the problems have been solved eventually. New experience and knowledge was gained during the process and effort trying to find the related solution for those problems. The following sections are some of the major problems that arose during the development process.

8.3.1 Lack of knowledge and experience in web-based programming

The web-based application development is relatively complicated compare to traditional stand-alone application. The programming of executable codes, structure designing is also required. In order to develop the system, research and study on related material have been carried out. Most of the researches found have provided assistance in solving various problems. Besides, discussion among friends and group members is another helpful alternative in order to solve the problem.

8.3.2 Lack of knowledge and experience in database server

Dealing with database server to store and manipulate the entire data of system comes across a lot of problem. The method of administer and manipulate the database is rather different as compare to non-database server program. A lot of time has been taken in learning the database server concept. Surfing Internet, reading up the related material and discussing with group member and friends did help in overcome the problem.

8.3.3 Insufficient development time

There are too many holidays for this semester, which had due to unavailability the faculty's lab to develop the system. As the send email module must have to be done in faculty's lab to try its server linking. In addition, more time was spent in learning, trying and searching solutions for the problem arose during the development of the system as unfamiliar with the development tools. As a result, some of the advance functions cannot be done within the expected time frame.

8.4 System Strength

CLT has developed in attaining its major objectives as well. Several strengths of the tool will be described in detail in the following sections.

8.4.1 Attractive and user-friendly interface

CLT is developed using the standard Windows application interface, which is both system efficient and user friendly. Standard GUI control objects, such as buttons, text box, scroll bar, radio, WIMP (Window, Icon, Menu and Pointing devices) are provided to allow new users to quickly understand the use of the tool. Animation on the main

page with simple text introduction of collaborative learning tool make the system interface become more attractive and eye-catching besides increase the curiosity to join in as new member of CLT.

8.4.2 Wide-accessibility

CLT is an online tool that allows teacher and student learning in cooperative way through Internet. Users can access the system virtually from anytime and anywhere in the world. The basic tool needed on the client-side PC is just Internet Explorer 4.0 or above, which is already preinstalled with Windows 98/2000 and XP professional.

8.4.3 Highly integrated module

All modules and sub-modules in CLT are highly integrated, where data change and updates in any module can be detected and update to all other linked modules. This can reduce data entry and management time besides keeping the concept consistency.

8.4.4 Authorization and Authentication

Each authorized user must have a login id and password that allow them to access the tool. The user should logging in each time before they enter and start using the tool. The tool will verify and validate the status of the user.

8.4.5 Reliability and Accuracy

CLT is a reliable tool because of the error tolerance provided by the tool. For any error-prone such as duplication of login id, missing require fields, invalid format or length of

required fields and etc, an error message for each process will be prompted stating the particular mistake. This can assure the accuracy of the data input by the user.

8.5 System Limitation

Due to time constraints, CLT does not include all the advanced functions that a tool should have. These limitations can be addressed in future enhancements.

8.5.1 Data Printing Function

CLT has not provided users, especially the instructors to print the data or information of the course contents or students' information in details from its applications. Some hardcopy might be useful for the instructor to keep record of the past, present and future of this related information.

8.5.2 Too Many Data Input Needed

The instructor, who has the power of full-control and full-manage of the whole collaborative learning process, might suffer to input all the data needed. The discussion could not be started whenever there are no data of course and students records.

8.5.3 Calendar Viewer

The calendar is just specified for the instructor to check their schedule to avoid timing crash for more than one activity. This functions is not enough advanced for the instructor to rearrange their schedule and make an appointment to the date accordingly. In addition, this module is not available for another user of the tool (student module).

8.5.4 Only One Recipient for Each Mail

For the send email module, only one recipient allowed for each email being sent. This might cumbersome the instructor if he/she wish to send email to whole course students with same contents.

8.6 Future Enhancement

System limitation should be addressed to enhance the functionality and features of CLT in future. The current version of the CLT can be updated with some enhancement features, which will discussed in the following sections.

8.6.1 Provide More Advance Functionality

More advance functions should be added to the CLT system to enhance its usability. There are including printing the needed data as store records for instructor, advance appointment making for the calendar module and allow sending email to more than one recipient in once. Besides, student should also be provided a calendar module so that they can learn how to arrange their schedule and time as well.

8.6.2 Automation Tool for Data Inputting

Most of the data input required instructor to key in one-by-one. This might due to more error-prone and cumbersome the instructor. CLT should provide an automation tool where some of the data such as student id or course id can be generated by tool itself instead of require the user input. This can save time and job of the instructor besides reducing the error-prone that is made by the user.

8.6.3 Encrypted the Database

All the data and information are stored in a plain text form rather than in an encrypted format. Hence, that is a need for the database to be encrypted in order to enhance the security of the CLT tool.

8.7 Conclusion

In conclusion, this project (CLT) has been successfully met it objective in developing a web-based collaborative learning tool. CLT integrates all its modules to provide a highly effective management solution for teachers to organize collaborative learning with their students, which is considered a tedious work in manual form. However, there are still some limitations in this tool, which need to be enhanced in the near future in order to transform its status to more advance tool.

Throughout the development of CLT, a lot of valuable knowledge and experience was gained. There are including the fundamental of web-based programming language in ASP.NET and VB.NET, configuration and manage the windows XP, web server (IIS) and SQL Server. Nevertheless, more importantly is the process of the project management and development has leaded me toward how to work in group and communicating among team members and project target users to plan, proceed and success a project.

Bibliography

[1] Ramsden, P. (1992) *Learning to Teach in Higher Education*, Routledge, London.

[2] Holyfield, S. & Liber, O. (1995) Using the Internet for Teaching: Using the World Wide Web for the Management of On-line Learning Resources in *Active Learning*, J. Darby & J. Martin (eds), July, 2:30-33.

[3] Green, D. G. & Jeffries, D. (1996) The New South Wales Higher School Certificate On-Line Project in Shaping the Wisdom of Oz, in *12th Annual Conference of the New South Wales Computer Education Group*, J. Attwood (ed.), NSW Computer Education Group Ltd, Sydney.

[4] Pickering, J. (1995) Using the Internet for Teaching: Teaching on the Internet is Learning in *Active Learning*, J. Darby & J. Martin (eds), July, 2:9-12.

[5] Dyer, B. (1995) Preparing for the 21st Century, *Innovations in Education and Training International*, 32(3):269-277.

[6] Blackboard Inc. (1997 – 2003). "Transforming the Internet into a powerful environment for the education experience". http://company.blackboard.com/. 23 July 2003.